Autocorrel I: A Neural Network Based Network Event Correlation Approach

Nathalie Japkowicz

Reuben Smith

School of Information Technology and Engineering
University of Ottawa
Ottawa Ontario

Contract Number: W7714-3-08710

Contract Scientific Authority: Maxwell Dondo

The scientific or technical validity of this Contract Report is entirely the responsibility of the contractor and the contents do not necessarily have the approval or endorsement of Defence R&D Canada.

DEFENCE R&D CANADA - OTTAWA

Contractor Report

DRDC Ottawa CR 2005-030

May 2005

© Her Majesty the Queen as represented by the Minister of National Defence, 2005

© Sa majesté la reine, représentée par le ministre de la Défense nationale, 2005

Abstract

Network event correlation is the process where correlations between network events are discovered and reported. Network intrusion detection analysts who have capable event correlation software at their disposal are more effective because the software can give an intrusion analyst a broader view of the threats posed to their system. The event correlation information is used by a network administrator to deduce the true relationship between individual network events. The autoassociator is ideally suited to the task of network event correlation. The autoassociator is a specialized piece of neural network architecture that can be used to cluster numerically similar data instances. We use the autoassociator to build prototype software to cluster network alerts generated by a Snort intrusion detection system, and discuss how the results are significant, and how they can be applied to other types of network events.

This page intentionally left blank.

Executive summary

Network event correlation is the process where correlations between network events are discovered and reported. If network events can be successfully correlated, the information can be used to help a network administrator or intrusion detection analyst. Clustering network events is an important task because it allows the administrator or analyst to analyze a set of related events as one, rather than deduce the relationship between individual events by hand, which takes more time. From the perspective of the intrusion detection analyst, such a system can save time by allowing the analyst to discount entire sets of alerts, if they fall into a set of alerts known to be insignificant.

The autoassociator is ideally suited to the task of network event correlation. The autoassociator is suited to this task because, unlike many machine learning algorithms, the autoassociator requires no initial set of data created and processed by humans. The autoassociator is a specialized piece of neural network architecture which can be used to cluster numerically similar data instances. We use this scheme to cluster network alerts generated by a Snort intrusion detection system, as a demonstration of the potential of this system.

The autoassociator is a neural network, a type of machine learning algorithm, which inputs a number of data instances and learns trends common to these data instances. Using these trends, the autoassociator can output a number called the reconstruction error, which is used to cluster like data instances together. Once like data instances are clustered with some reasonable degree of certainty, the intrusion detection analyst may use the information as he sees fit.

In this report, we present the results we obtained from this scheme by analyzing the output of the autoassociator. We were able to cluster together related Snort alerts quite well, as we demonstrate. The system is able to cluster together similar alerts, but we note that it sometimes also clusters together dissimilar alerts because of a failing in the system. Dissimilar alerts are clustered together because of the way we map the many autoassociator outputs to the reconstruction error range.

Nathalie Japkowicz, Reuben Smith; 2005; Autocorrel I: A Neural Network Based Network Event Correlation Approach; DRDC Ottawa CR 2005-030 ; DEFENCE R&D CANADA - OTTAWA.

This page intentionally left blank.

Table of contents

Abstract i

Executive summary iii

Table of contents v

1 Introduction 1

2 Background 3

- 2.1 Intrusion Detection Systems 3
  - 2.1.1 Existing Intrusion Detection Systems 4
  - 2.1.2 Anomaly-based Detection Research 5
  - 2.1.3 Previous Event Correlation Research 5
- 2.2 Artificial Neural Networks 6
  - 2.2.1 The Neuron 7
  - 2.2.2 The Activation Function 7
  - 2.2.3 Multi-Layer ANNs 9
  - 2.2.4 ANN Training 10
  - 2.2.5 Training Rules 11
  - 2.2.6 The Autoassociator 13

3 Our Model 15

- 3.1 The Data 15
  - 3.1.1 Gathering and Selecting 15
  - 3.1.2 Attribute Selection 16
  - 3.1.3 Formatting 17
  - 3.1.4 Formatting Example 18
- 3.2 Training the Autoassociator 20

3.3 Clustering Autoassociator Output 21

4 Results Analysis 23

4.1 Analysis of Example Clustering 23

4.2 Numerical Results 26

4.3 Errors Made 27

5 Conclusions and Future Work Recommendations 28

References 29

Annexes 32

A Acronyms and Abbreviations 32

B Full Output for DARPA Data 33

C Progress Reports 47

C.1 January Report 47

C.1.1 Problem Overview 47

C.1.2 Data 48

C.1.3 Potential Areas of Investigation 49

C.1.4 Outline Project Timeline 51

C.2 February Report 52

C.2.1 Overview of Report 52

C.2.2 Data Processing 52

C.2.3 Data Sets 55

C.2.4 Initial Analysis 56

C.3 March Report 60

C.3.1 Overview of Report 60

C.3.2 Clustering With The Autoassociator 60

C.3.3 Comparison of Clusterers 62

C.3.4 Clustering Example 63

C.3.5 Event Cluster Correlation 66

C.3.6 Group Correlations Implementation 67

C.4 April Report 70

C.4.1 Overview of Report 70

C.4.2 The Algorithm 71

C.4.3 Analysis of Output 72

C.4.4 Summary of Output Analysis 75

C.5 Appendix to April Report 77

C.6 DARPA Test Data 112

C.7 Incidents.org Data 119

1 Introduction

Our goal in this research is to aid the intrusion detection analyst by speeding up their job of sorting through network events. We hoped that by discovering meaningful correlations of separate network events, we would help the analyst prioritize the alerts they might consider a threat. To this end, we consider a system to cluster network events so that the analyst might be able to analyze multiple network events in a more efficient manner.

Neural networks are a type of machine learning architecture useful in storing an abstracted form of large amounts of data in a novel way. Instead of storing all of the data unfiltered, neural networks are designed to store the trends of the data rather than the actual data. The trends represented in a neural network are used to classify new data or to analyze new data.

The autoassociator is a specialized piece of neural network architecture which can be used to process data in an unsupervised way. Unlike conventional neural networks, the autoassociator need not be used explicitly to classify data; it may be used to recognize previously seen data and to assign a numerical rank to the data. The autoassociator is elegantly suited to our task of clustering network events. The autoassociator's ability to learn from unstructured, unprocessed data makes it easy to implement and test with for the network event data. The performance of our end system lies in the validity of the correlations we find, so we must show that the correlations we find are useful. We are able to show the usefulness of the discovered correlations because it is easy to test with the autoassociator. Neural networks in their general form and the autoassociator are explained in technical detail in Section 2.2.

The system we have produced uses the autoassociator to find correlations in the network events we examine. To feed network event data to the autoassociator style of neural network, we must encode the network event as a string of numbers from which the autoassociator can learn numerical trends. We show how we do this encoding of network events for the prototype system in Section 3.1.3, and we discuss the details of how the autoassociator learns from this data in Section 3.2. As a last step in the system, we must interpret the autoassociator output, so we discuss the algorithm we used to do this in Section 3.3.

For a prototype of the system we propose, we use the output of an intrusion detection system (IDS) as the set of networks events in which we wish to find correlations. We made this choice of data for a few reasons: (1) there is publicly accessible data, which can be fed into a correlation system, at incidents.org [1] and from DARPA [2] – publicly accessible routers logs, firewall logs, or the like are rare; (2) the output we use from the IDS is essentially filtered, so the data instances we're dealing with are more interesting than the data before it was filtered through the IDS; and (3) we felt that finding correlations in only this data set would give sufficient proof of the concept, and that we did not sacrifice generality by using data from a single source of events in the analysis of our correlation system. We give some background on intrusion detection systems in Section 2.1.

Finally, we demonstrate how this system will work by applying the system to the 1999 DARPA data set [2] for IDSs. Because of the nature of our problem, we use this data set in different way than other projects that use the data set. We use Snort [3] to extract alerts from the supplied raw TCPDump [4] format data, then we run our correlation system on the alerts, regardless of whether the alerts are false alarms. The goal of this research is not to reduce the number of false alarms, but rather to find correlations between the given alerts. By finding these correlations, we may be helping the IDS analyst determine false alarms, because the IDS analyst will be able to discount a set of alerts quickly, but this is not an explicit goal. We demonstrate the system on this data set in Section 4.

Because we use the 1999 DARPA data set [2] only as a source of network events rather than also a way to measure the accuracy of our system (as other projects which use the DARPA data set have), the accuracy results we present are not comparable to the results of other projects that test with the same data set. We do not use the testing data provided in DARPA data set because we are not testing our accuracy against the data set; our accuracy measures reflect the degree to which our outputted clusters are well-formed. If there are a high number of false positives and low number of true positives in the Snort output that we use as our input, it does not hamper our correlation analysis effort; we are able to discuss the validity of a cluster of alerts, regardless of whether the individual alerts are valid or not.

As well, please note that the bulk of this final report is summary of the work we've completed in the previous reports for this research project. To see the reasons we did not consider certain choices, or why we arrived at our conclusions, it may be useful to read these previous reports as background. These previous reports are available as appendices, in Annex C.

2 Background

The IP protocol is described in sufficient detail by Stevens [5]. The Stevens book is necessary reading before advancing to the weightier network intrusion detection system material by Northcutt [6], relating directly to our task. We talk about the current state of IDS research in Section 2.1.

We give detailed background information on artificial neural networks in Section 2.2, and we talk about a specific, relevant type of neural network, the autoassociator, in Section 2.2.6.

2.1 Intrusion Detection Systems

Intrusion detection systems (IDSs) are systems which report packets or connections that are considered suspicious by some set of rules, or by statistical analysis against some pre-defined set of normal traffic. They exist to solve the problem of attackers penetrating a target system of an unknowing system administrator. IDSs can be deployed inside or outside the network's firewalls depending on the type and level of detail of an attack that an administrator wishes to see. IDSs are generally considered to be low-level tools in analyzing network traffic because of the vast number of possible attacks they report, and because they do not indicate to the analyst the context of the supposed attack.

Richard Steven's TCP Illustrated [5] volumes are a good starting point for those interested in the raw workings of the TCP protocol. Richard Bejtlich [7] wrote a good introduction on what is expected of an intrusion detection analyst when analyzing traffic, and on how intrusion detection systems can give a false sense of security to those who rely too heavily on them. Bejtlich's paper advocates the use of TCPDump [4] for the intrusion detection analyst to obtain a layer lower than the output presented at the IDS console.

Steven Northcutt wrote a very good book on intrusion detection called Network Intrusion Detection: An Analyst's Handbook [6], where he presents the basic problems motivating intrusion detection, and the most applicable solutions to the problems. He also provides information on how most IDS vendors deal with the problem of intrusion alert correlation. We sometimes used his ideas as a basis for decisions we made in the design of our system, though Northcutt does not treat the correlation problem in the light of machine learning.

We use the terms *intrusion detection* and *network intrusion detection* interchangeably in this report, although they have not always been synonymous in previous research by other authors. Historically, *intrusion detection* implies host-based or network-based instrusion detection, and *network intrusion detection* implies only the latter. The former tends to refer to attacks where a single host is targetted, and the latter tends to refer to attacks which concern any component of the network.

2.1.1 Existing Intrusion Detection Systems

EMERALD [8] is a distributed, scalable, hierarchal, customizable network intrusion monitor. It can compose other intrusion detect systems, to correlate threat warnings between different systems. At the time of writing this paper, the updaters of this system had not released their correlation unit to the public. EMERALD explicitly divides statistical analysis (anomaly detection) from signature-based analysis (misuse detection), and recombines the results at a higher level.

Snort [3] is a lightweight, open source intrusion detection system, which doesn't do TCP or IP defragmentation. We don't consider this defragmentation feature important for our research, because if IP packets are fragmented to such granularity that their malicious intent is hidden, then any anomaly-based system will see these packets as anomalous. (Snort flags these packets, even though it's ruled-based, since this rule is turned on by default.)

ACID [9] is a web-based intrusion detection console that offers only very simplistic correlation abilities, but which can interoperate easily with Snort. In ACID, the IDS analyst can correlate alerts by source or destination TCP port, or source or destination IP address. This single-variable analysis leads to problems when trying to detect attacks such as distributed denial of service attacks.

Shadow [10] is an IDS created by Stephen Northcutt et al. In his book [6], Northcutt mentions alert correlation, but only in a capacity similar to that found in ACID. He mentions correlation by time, as well as by the other single-variable attributes listed for ACID.

NetSTAT [11] is an IDS which focusses on *network intrusion detection* rather than *host intrusion detection*. Similar to EMERALD, NetSTAT is scalable and composible.

QuidSCOR [12] is an open-source IDS, though it requires a subscription from its publisher, Qualys Inc., to work correctly. It works with IDSs such as Snort to correlate alerts against Common Vulnerability and Exposure (CVE) information published by Mitre.org.

Intellitactics/NSM [13] and Network Flight Recorder (NFR) [14] are other IDS systems that purport to have correlation abilities, but their software isn't free to download, so their claims were not investigated more thoroughly.

2.1.2 Anomaly-based Detection Research

Anomaly-based intrusion detection research typically trains its systems on labelled network traffic, which contains no intrusions or abnormal traffic that are not labelled as such for the learning algorithm which processes the traffic. These systems are then tested against traffic containing a small proportion of intrusions to normal traffic. These systems exist in contrast to misuse detection systems, which typically employ hand-coded rules that must be updated continually to reflect the most current attacks being perpetrated against the network.

"A Multiple Model Cost-Sensitive Approach for Intrusion Detection" by Wei Fan, Wenke Lee, Salvatore J. Stolfo, and Matthew Miller [15] tries to better classify network intrusions by seeing them as anomalies to normal network traffic. Rather than detect intrusions by matching incoming packets to an administrator-created signature, Lee and Stolfo's work [16] in combining data mining with intrusion detection trains a machine learning algorithm to recognize non-intrusion traffic so that it can be discerned from intrusion traffic. New intrusions will not be classified as such in signature-based IDSs, so anomaly-based systems remedy this problem by design. Lee and Stolfo [16] explore both signature-based detection (also known as misuse-detection) and anomaly detection, and compare the results. In Fan et al. [15], the authors expand on this work by showing the system can be optimized with respect to operational costs of the IDS. In [16] and [15], the authors use the RIPPER [17] algorithm for classification of data from the DARPA [2] and KDD99-Cup [18] data sets.

In Eskin et al. [19], the authors build on the work in [16] and [15] to extend the work to unlabelled data – this refers to data that doesn't have each packet labelled as *normal* or *anomaly*. This new method leads to an unsupervised anomaly detection algorithm which can be deployed with minimal training effort. Eskin et al. [19] also generalize their approach so that any unsupervised machine learning algorithm can be used; they compare the results of a cluster-based estimation algorithm, the K-nearest neighbour algorithm, and the one class SVM algorithm. They find that the one class SVM algorithm wins out if higher false alarm rates are allowed, and they use ROC curves [20] to present their results.

2.1.3 Previous Event Correlation Research

In terms of network event correlation, few papers have been written. We review the ideas in three papers as an overview of existing research.

First, Ning and Cui [21] take an idealist view of network event correlation. Ning and Cui recognize that many existing IDSs tend to detect low-level events without being able to relate these events to the broader plan of the attacker. Their goal is to create hierarchies of alerts using prerequisites and consequences for each type of alert. They attain this goal using a rule-based system (notably rather than a learned system) that assigns prerequisites and consequences to each type of alert, which in turn allows the analyst to quickly see the possible consequences of the most mundane of alerts. Their system runs on the output of existing IDS systems. They demonstrate the usefulness of their system by showing how their system can significantly reduce the number of false alarms reported by an IDS while negligibly reducing the valid alarms reported by the IDS. Their tool is most logically used to run as an offline forensic tool for mining old stored alerts after a new vulnerability is found.

Debar and Wespi [22] present an aggregation and correlation component (ACC) for an IDS. They use this correlation component to flag alerts as an original, a consequence, or a double alert. In this research, they present rule-based methods for attaining this goal. They concern themselves with implementation issues such as integrating their ACC with existing IDSs – namely the Tivoli Enterprise Console, and they suppose the use of the IDWG [23] format for their work – and they discuss other issues such as raising or lowering the priority of IDS alert to reflect the inferences of their ACC.

Valdes and Skinner [24] present a paper for using statistical techniques in the network event correlation problem. They present a system to fuse alerts together from heterogeneous network sensors into single, more easily understood alerts. They control how loosely their fusing algorithm works with a single system parameter. Their system is criticized by Ning et al. [21] as not being an effective correlation effort because often the groups of alerts that the system should find are not tied together by underlying numerical similarities.

2.2 Artificial Neural Networks

The area of artificial neural networks (ANNs) have been a subject of intense research lately. As a result, the field of ANNs is now very broad. In this section, we present a brief overview of ANNs.

ANN models attempt to emulate the human brain through the dense interconnection of simple computational elements called neurons [25]. Each neuron is linked to some of its neighbours through synaptic connections of varying strengths. Learning is accomplished by continuously adjusting these connection strengths (weights) until the overall network outputs the desired results. These weight adjustments are based on mathematical algorithms used in solving nonlinear optimization functions.

2.2.1 The Neuron

Similar to the biological nervous system, the basic computational element of an ANN is called the neuron or processing node. The neuron model is based on highly simplified considerations of the biological neuron. A simple node is shown in Figure 1, where N inputs are summed at the node. Each input u_i is connected to the processing node through the synaptic connections, which are represented by connection strengths called weights w_i. A bias term θ is also used at each node. The sum is fed through a transfer function f, called the activation function, to generate the output o. The signal flow is considered unidirectional as indicated by the arrows.

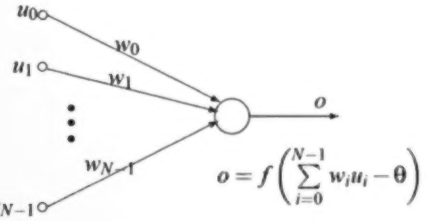


***Figure 1:** The basic neuron*

Although ANNs are constructed using this fundamental building block, there are significant differences in the architectures and driving fundamentals behind each ANN model.

2.2.2 The Activation Function

The activation function f plays a pivotal role in the functioning of the neuron. It determines the node output. As in Figure 1, the neuron output signal is given by:

$$o = f\left(\mathbf{w}^T \mathbf{u}\right) \tag{1}$$

where $\mathbf{w}$ is the weight vector defined as

$$\mathbf{w} \equiv \begin{bmatrix} w_1 & w_2 & \cdots & w_N \end{bmatrix}^T$$

and the input vector $\mathbf{u}$ is defined as

$$\mathbf{u} \equiv \begin{bmatrix} u_1 & u_2 & \cdots & u_N \end{bmatrix}^T$$

There are many different types of activation functions f to choose from, depending on the application [25–27]. Some of the commonly used activation functions are shown in Figure 2. These activation functions are the *hard-limiter*, the *threshold logic*, and the *sigmoid*. Since real applications are usually modeled as continuous functions, the most commonly used continuous activation function is the sigmoid.

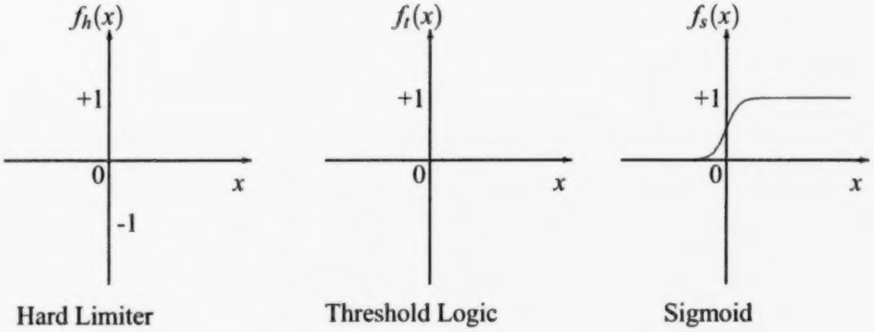


***Figure 2:** Activation functions*

Activation functions may be either unipolar, for positive output, or bipolar for output that may be positive or negative. For example, the bipolar sigmoidal activation function is defined as:

$$f(x) \equiv \frac{2}{1+\exp^{-\lambda x}} - 1 \tag{2}$$

and the unipolar sigmoidal activation function is defined as

$$f(x) \equiv \frac{1}{1+\exp^{-\lambda x}} \tag{3}$$

where λ is a constant.

A special case of an ANN is a single node based on the neuron model shown in Figure 1 and is called a *perceptron* after the work of Rosenblatt [25]. A perceptron consists of one or more neurons. If a continuous activation function is used, the neuron model is known as a *continuous perceptron*. A continuous perceptron is capable of classifying *linearly separable* classes of data of the form $a\mathbf{x}+b$. Multiple nodes in this format form a single layer multi-node ANN capable of classifying linearly separable data patterns.

2.2.3 Multi-Layer ANNs

To emulate massively interconnected biological systems, ANNs have to be similarly interconnected. ANNs are the simple clustering of primitive artificial neurons. This clustering occurs by creating layers of neurons which are connected to one another. Figure 3 shows a multi-layer perceptron. An input layer interfaces with the outside world to receive inputs and an output layer provides the outside world with the network's outputs. The rest of the neurons are hidden from view, and are called *hidden layers*.

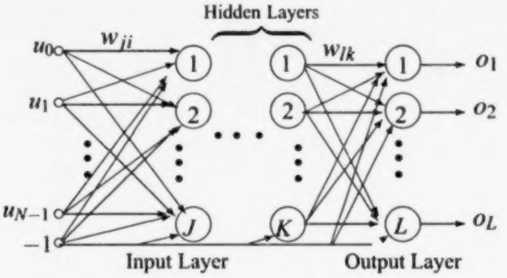


Figure 3: *Multi-layer perceptron*

The objective of using a multi-layer perceptron is to be able to classify patterns that linear classifiers (single layer ANNs) are incapable of classifying. The most important attribute of multi-layer ANNs is that they can learn to classify a problem of any complexity. The biggest challenge is usually in deciding the number of hidden layers in an ANN.

Zurada [27] gives an extensive discussion on the design of the number and size of hidden layers in a given architecture; nevertheless, trial and error methods have been widely used. If the number of hidden layers is too large, the ANN architecture will have problems generalizing; it will simply memorize the training set, making it useless for use with new data sets.

Inter-layer connections within an ANN architecture can take the following forms [27]:

- In a *fully-connected* ANN, each neuron on one layer is connected to every neuron on the next layer.
- In a *partially-connected* ANN, a neuron on one layer does not have to be connected to all neurons on the next layer.

If signal flow direction is taken into consideration, these two architectures can be further refined:

- In a *feedforward* ANN, the neurons on one layer send their output to the neurons in the next layer, but they do not receive any input back from the neurons in the next layer.
- In a *bi-directional* ANN, the neurons on one layer may send their output to the next layer or the preceding layer, and the subsequent layers may also do the same.
- In a *hierarchical* ANN connection, the neurons of a lower layer may only communicate with neurons on the next level of layers.
- In a *resonance-connected* ANN, the layers have bi-directional connections, and they can continue sending messages across the connections a number of times until previously defined conditions are achieved.

In more sophisticated ANN structures the neurons communicate among themselves within a layer, this is known as intra-layer connections. These take the following two forms:

- In fully- or partially-connected *recurrent* networks, neurons within a layer communicate their outputs to neurons within the layer. This is done a number of times before they are allowed to send their outputs on to another layer.
- In *on-center/off-surround* ANNs, a neuron within a layer has excitatory connections to itself and its neighbors, and has inhibitory connections to other neurons. The neurons exchange their output signals a number of times until a winner is found. The winner is allowed to update its and its members' weights.

The overall architecture of an ANN depends on the mappings required, the type of input patterns, and the learning rules to be used.

2.2.4 ANN Training

Similar to the brain, ANNs learn from experience by changing the ANN's connection weights until a solution is found. The learning ability of an ANN is determined by its architecture and by the algorithm chosen for training. The training methods [25] fall into broad categories:

- In *unsupervised training*, hidden neurons find an optimum operating point by themselves, without external influence.

- *Supervised training* requires that the network be given sample input and output patterns to learn. It is guided through the learning process until a satisfactory optimum operating point or a predefined threshold is reached. The most common training termination criteria is by setting a training threshold.

Backpropagation training is a form of supervised learning that has proven highly successful in training multi-layered ANNs. Information about errors is filtered back through the system and is used to adjust the connections between the layers, thus improving performance.

ANNs can be trained *on-line* or *off-line*. In off-line training algorithms, its weights do not change after the successful completion of the initial training. This is the most common training approach; especially in supervised training. In on-line or real time learning, weights continuously change when the system is in operation [27].

2.2.5 Training Rules

There is a wide variety of learning rules that are used with ANNs. Error minimization algorithms are used to determine the convergence levels when updating weights. In general, all ANN learning involves the iterative updating of the connection weights until the desired convergence is achieved. Most training algorithms start by initializing the weights to 0 or very small random numbers. This weight update is given by:

$$\mathbf{w}^{k+1} = \mathbf{w}^{k} - \Delta\mathbf{w}^{k} \tag{4}$$

Equation 4 is the ANN *general learning rule* [27]. The numerous learning rules, which are variations of this rule, only differ by the mathematical algorithms used to update the connection weights, or more specifically to calculate the value of $\Delta\mathbf{w}^{k}$ at each iteration k. Some of the common training rules are as follows:

- In the *Hebbian* rule [26,27], the connection weight update $\Delta\mathbf{w}^{k}$ is proportional to the neuron's output. This was the first ANN learning rule [25,28].
- The *perceptron* rule [25] updates the weights based on the difference between the desired output d and the actual neuron's response o.
- The *delta* learning rule [25,28] is based on the minimisation of the mean square error (MSE) as represented by the error function E, as shown in Equation 5.

$$\mathbf{w}^{k+1} = \mathbf{w}^{k} - \eta \nabla E\left(\mathbf{w}^{k}\right) \tag{5}$$

where η is a learning constant, and ∇E is the gradient of the error function E, defined by:

$$E_k = \frac{1}{2}\left(d^k - o^k\right)^2 \tag{6}$$

The objective is to iterate Equation 5 until the error E approaches zero (or a preset threshold value).

For an ANN with P training patterns, and K outputs, the *root-mean square error* (also known as the MSE [27]) is defined as:

$$E_{rms} = \frac{1}{PK}\sqrt{\sum_{p=1}^{P}\sum_{k=1}^{K}(d_{pk} - o_{pk})^2} \tag{7}$$

- The *Widrow-Hoff* [26,27] learning rule (sometimes called the *Least Mean Square* learning rule) is considered a special case of the delta learning rule in that the neuron output o is independent of the activation function f.

- The most widely used supervised training approach which is derived from the Widrow-Hoff algorithm is the *error backpropagation training algorithm*. As the name implies, the error $\Delta \mathrm{w}^k$ is propagated back into the previous layers. This is done one layer at a time, until the first layer is reached.

  Consider an ANN with one hidden layer, K outputs, J hidden nodes, I inputs, and P training patterns. The output layer weights are adjusted as follows:

$$w_{kj} = w_{kj} + \eta\delta_{ok}y_j, \quad \text{for k} = 1,\cdots,\text{K},\ \text{j} = 1,\cdots,\text{J} \tag{8}$$

  where η is a learning constant and the output error $\delta_{\mathbf{ok}}$ is given by

$$\delta_{ok} = \frac{1}{2}(d_k - o_k)(1 - o_k^2), \quad \text{for k} = 1,2,\cdots,\text{K} \tag{9}$$

  The weight update for the hidden layer is as follows:

$$w_{ji} = w_{ji} + \eta\delta_{yj}u_i, \quad \text{for k} = 1,\cdots,\text{K},\ \text{i} = 1,\cdots,\text{I} \tag{10}$$

  where the output error $\delta_{\mathbf{yj}}$ is given by

$$\delta_{yj} = \frac{1}{2}(1 - y_j^2)\sum_{k=1}^{K}\delta_{ok}w_{kj}, \quad \text{for j} = 1,2,\cdots,\text{J} \tag{11}$$

  The process is iteratively repeated until a preset threshold of the MSE (Equation 7) is achieved.

A good, longer treatment of neural networks in all their incarnations, see the book Neural Network Design by Hagan et al. [29].

2.2.6 The Autoassociator

Based on the work by Japkowicz [30], we extend the uses of the autoassociator to a new application. This new application is clustering alerts outputted from an intrusion detection system. We give the details of this application in Section 3.2.

The autoassociator is a piece of feedforward, fully-connected, multi-layer neural network architecture whose design is specialized for recognizing one class of training data rather than discriminating between training input of different classes, as other neural network architectures are designed to do. The autoassociator is known as an unsupervised machine learning algorithm because the classifier is trained on one class of data, so that it can recognize that class of data. Although the autoassociator is trained on only one class of data, it's tested on both classes of data in a binary learning problem (ie- two class problem).

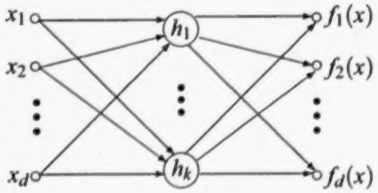


Figure 4: *The autoassociator*

In Figure 4, d is the number of attributes of the data items in the data set. $x_1,\ldots,x_d$ represent the input values of the attributes for a data item, $f_1(x),\ldots,f_d(x)$, represent the reconstructed output values of the autoassociator, and $h_1,\ldots,h_k$ represent one layer of k hidden units in Figure 4. The potential of the autoassociator to internally represent data trends often lies in the number of hidden units and design of the hidden layer, so these parameters should be carefully tailored to the problem.

The most clear difference between the design of the autoassociator in Figure 4 and the multi-layer perceptron is that the autoassociator has d output neurons, where the multi-layer perceptron has only 1 output neuron. This difference is crucial.

The reconstruction error for a data item is computed using the $f_1(x)$, $f_2(x)$, $\ldots$, $f_d(x)$ autoassociator output values from Figure 4. The formula for computing the reconstruction error is:

$$E_x = \sum_{i=1}^{d} [x_i - f_i(x)]^2 \tag{12}$$

where x_i is the value of the i^{th} attribute of data item x and $f_i(x)$ is the value of the i^{th} output of the neural network with input x. After training, the autoassociator is able to recognize the class represented in the training data because data items from the training class will have low reconstruction error values when compared to that of data items from unseen classes. The theory holds that data items from classes without representative training examples will have higher reconstruction errors.

A good treatment of using the autoassociator for this type of classification is [30]. This paper explains more of the theory behind the operation of the autoassociator, as well as applying this style of neural network in a different way from what we present here.

3 Our Model

The prototype system we have produced uses the autoassociator to find correlations in the network events we examine. The network events we're concerned with for our experiments are Snort-generated IDS alerts. We hypothesize that if our system works correctly for this type of network event, it will work correctly for other types of network events, such as router errors or firewall warnings.

Firstly, we used Snort on the DARPA data set to produce a large number of textual alerts. We then converted these alerts into numerical representations for the autoassociator code that we'd created using Matlab's Neural Network Toolbox [31], in the way described in Section 3.1. We trained the autoassociator on 10,000 alerts, then used the trained autoassociator to generate reconstruction error values for 100 alerts. The details on how we use the autoassociator for this project are in Section 3.2.

After producing these reconstruction error values, we clustered them using the algorithm we present in Section 3.3. We analyze the results of this clustering in Section 4.

3.1 The Data

This data is first read using Snort [3], then output as textual alert data to Perl [32] scripts that can understand the Snort alerts and can format the data for use with the autoassociator algorithm.

3.1.1 Gathering and Selecting

We tested our system using two data sets, but we only report the results of testing one of the data sets in this final report. For the results of our system on the incidents.org [1] data set, please see the April report in Section C.4. We originally used the incidents.org data set because the results we received from this data set were interesting, and were a good demonstration of our fledgling system.

For this report, we demonstrate our system using 1999 DARPA [2] data set. We chose to use the 1999 DARPA data set because it is a published data set that is widely used in the field of IDS research. The incidents.org data set was not artificially generated, and it produced interesting results, but the DARPA data set is sufficient to demonstrate the system.

To gather the data instances for the testing and training classes, we took the TCP-Dump [4] data from week 5 of the DARPA data set and pushed it through Snort.

We saved all of the alerts that Snort generated. We chose the fifth week of the DARPA data set because the first week was the training part of the data set and it was designed to contain no interesting alerts.

We then reserved the first 1000 alerts generated in this week for testing, and we reserved the next 10,000 alerts for training. We recognize that a proper experiment would be to train on the first 10,000 alerts, then test on the remaining alerts, since alerts appear earlier in the Snort output if they were encountered earlier. Theoretically, testing on alerts from before the training period could invalid the test since it's not a realistic scenario for a real-world deployment – that is, in the real world the classifier will always be trained on previously seen or labelled data, then deployed for future data – but the realistic concern that the sample of alerts at the beginning were more diverse and interesting trumped this theoretical concern.

3.1.2 Attribute Selection

We have 40 attributes to characterize each Snort alert. The attributes we used were inferred directly from the textual Snort alert. Our method of attribute selection mostly consisted of removing attributes rather than creating new ones. We decided not to use any of the information created in Snort's processing. For instance, we do not use the Snort attack priority level or Snort's labelling of the attack because this information cannot be found in the raw IP packet. We wanted to make our prototype system as general as possible, so we showed how using only protocol attributes can still allow for very good clustering with the autoassociator.

We did not include the IP source address or IP destination address protocol fields in the list of classification attributes because we noticed that these attributes tended to impede correct clustering. We also reason that if we wish to cluster a number packets together that make up a distributed denial of service (DDOS) or similar attack, the IP address fields could only impede classification since they are easily forged.

We kept the TCP source and destination port attributes because we found they were useful in clustering types of attacks on particular TCP services.

We considered removing the TCP sequence and TCP acknowledgement numbers from the list of selected attributes, since they can impede classification in some cases, but we found they also helped classification in some cases (especially in the incidents.org data set). We chose to remove these attributes for the DARPA data set clustering, but keep them for the incidents.org clustering.

We tested and considered the other attributes individually, but we decided to keep them all because they all helped classification. In the end, we had 40 attributes that we passed to the autoassociator for classification. These attributes are listed in Figure 5.

Feature	Feature	Feature	Feature
portSrc	portDest	ipIsIcmpProtocol	ipIsIgmpProtocol
ipIsTcpProtocol	ipIsUdpProtocol	ipLen	ipDgmLen
ipId	ipTos	ipTtl	ipOptLsrr
ipPacketDefrag	ipReserveBit	ipMiniFrag	ipFragOffset
ipFragSize	icmpCode	icmpId	icmpSeq
icmpType	tcpAckNum	tcpFlag1	tcpFlag2
tcpFlagUrg	tcpFlagAck	tcpFlagPsh	tcpFlagRst
tcpFlagSyn	tcpFlagFin	tcpLen	tcpSeqNum
tcpWinNum	tcpUrgPtr	tcpOptMss	tcpOptNopCount
tcpOptSackOk	tcpOptTs1	tcpOptTs2	tcpOptWs
tcpHeaderTrunc	udpLen		

***Figure 5:** Selected Attributes*

We did not consider any more advanced, automatic feature selection algorithms for this research project because we found that the autoassociator is quite robust in ignoring attributes which do not add to correct classification.

3.1.3 Formatting

All of the 40 attributes we selected were quite easy to convert to numerical data. For instance, if the TCP Ack flag was set in the TCP header (if there was a TCP header), the value of the tcpAckFlag attribute was set to 1, and if there was no TCP header or the flag was clear, the tcpAckFlag attribute set to 0. The only slightly more complicated conversion took place in converting IP addresses to numbers, and we didn't need to do this after we dropped the IP address attributes (see Section 3.1.2).

We normalize the data using the common method (see [33]) of determining a linear map from the training data then applying it to both the training data and the testing data. To be more precise, for each attribute of the training data, we store the smallest (*min*) and largest (*max*) values of the data items, then linearly map the data of that attribute from the computed range [*min*,*max*] to the interval [0,1]. For each attribute, we map the testing data using the values determined from the training data to the range [0,1] as well, allowing data to be mapped outside this range if testing data contains values smaller or larger than previously seen.

The necessity for normalization is explained succintly in a paper on using support vector machines (SVMs) [34]. Lin gives a concise overview of why scaling is important – basically, so that one feature doesn't overwhelm other ones – and why it's okay to have testing data outside of the chosen scaling interval. The precise range for the testing cannot be known, by the definition of the testing data, and to allow for a realistic representation of the testing data, the model must acknowledge that training data rarely perfectly predicts testing data.

Some features of note are the `ipIsIcmpProtocol`, `ipIsIgmpProtocol`, `ipIsTcpProtocol`, and `ipIsUdpProtocol` binary features. If a data item has the value one for the `ipIsTcpProtocol` feature, this indicates that the Snort alert signifies an IP packet containing a TCP payload. Since the web server in this alert listens on a TCP port, this feature has value 1 for this data item. A Snort alert for a non-TCP IP packet will have `ipIsTcpProtocol` set to zero. Only one of the `ipIs*Protocol` features will be set for a given data item. We created these features because we imagined that it would be possible for two packets of the same protocol to be essentially identical in all respects except for the IP protocol field.

If the packet does not contain a TCP payload, the data instance representing the alert has value -1 for the `portSrc` and `portDest` features, to indicate that a non-present value was observed.

3.1.4 Formatting Example

To illustrate precisely how the transformation is made from a textual alert to a data item, and what information is encoded, we present an example. (Although the data items we output can be normalized, we present a non-normalized data item here for readability.) The following Snort-generated alert was processed with our scripts to generate a row of numbers, where each number corresponds to a predetermined feature. The alert below was generated by Snort after a possible attacker tried to access the UNIX `telnet` service. It's possible that this alert could have been generated by a valid user, but it's also possible that this alert could have been generated by an attacker trying to penetrate the system. If this second hypothesis were true, it is likely that you would find many `TELNET login incorrect` errors close to this one targetting the same machine.

```
[**] [1:718:7] TELNET login incorrect [**]
04/05-09:38:26.936288 172.16.114.50:23 -> 172.16.114.168:10332
TCP TTL:64 TOS:0x10 ID:2014 IpLen:20 DgmLen:66 DF
**AP*** Seq:0x801D2FDC Ack:0x6EDB5C7F Win:0x7C00 TcpLen:20
```

The Perl scripts we've created encode this alert information as predetermined features. We've created a table in Figure 6 for all the features of the data item. As an example of how to read Figure 6, in the data below, the number 23 is the TCP source port (`portSrc`) given in the alert.

Feature	Value	Feature	Value
portSrc	23	portDest	10332
ipIsIcmpProtocol	0	ipIsIgmpProtocol	0
ipIsTcpProtocol	1	ipIsUdpProtocol	0
ipLen	20	ipDgmLen	66
ipId	2014	ipTos	16
ipTtl	64	ipOptLsrr	0
ipPacketDefrag	1	ipReserveBit	0
ipMiniFrag	0	ipFragOffset	0
ipFragSize	0	icmpCode	0
icmpId	0	icmpSeq	0
icmpType	0	tcpFlag1	0
tcpFlag2	0	tcpFlagUrg	0
tcpFlagAck	1	tcpFlagPsh	1
tcpFlagRst	0	tcpFlagSyn	0
tcpFlagFin	0	tcpLen	20
tcpWinNum	31744	tcpUrgPtr	0
tcpOptMss	0	tcpOptNopCount	0
tcpOptSackOk	0	tcpOptTs1	0
tcpOptTs2	0	tcpOptWs	0
tcpHeaderTrunc	0	udpLen	0

Figure 6: *An Encoded Alert*

As you can see from the Snort alert, the TCP header in this packet has only the Ack and Push TCP flags set. (The string `***AP***` indicates this.) This information is encoded in the `tcpFlag*` group of features. `tcpFlagAck` and `tcpFlagPsh` are set to 1 and the rest of the flags are set to 0.

The `portSrc` and `portDest` features represent the TCP source and destination ports for a Snort alert. The values of the `portSrc` and `portDest` features are taken directly from the alert, as you can see above. The value for `portSrc` and `portDest` are 23 and 10332 respectively. The TCP source and destination port features are important in finding good correlations because they are often the only obvious indicator to which protocol the alert relates.

3.2 Training the Autoassociator

Machine learning is used in this project to cluster numerically similar network events. We will show how the autoassociator, an elegant neural network design, can be used for exactly this task. There is a post-processing step required to interpret the neural network output, which we'll discuss in the next section, but the autoassociator is used in the same way as is described in Section 2.2.6. The difference between the way we use the autoassociator and how it is used in "Supervised Versus Unsupervised Binary-Learning by Feedforward Neural Networks" by Nathalie Japkowicz [30] is in what type of data we input to the neural network and how we process the output from the network.

In our task, we use the autoassociator for clustering, rather than for binary classification. Where [30] found a cutoff in the reconstruction error values to separate two classes of data, we group the reconstruction error values to form clusters. The difference in reconstruction error of two given data is positively correlated to the general numerical difference of those data. In our task, if the numerical representation of two network events is similar, the autoassociator will produce similar reconstruction errors. The converse is often true, but not always true: numerically dissimilar network events often produce dissimilar reconstruction errors, but not always. The reason this is the case is because the reconstruction error formula maps the 40 autoassociator outputs to the one-dimensional range of error reconstruction values. Because of this 40-to-1 mapping, there is a lot of information lost, so numerically different errors can be mapped to the same reconstruction error. This happens occasionally, and fixing this problem is a subject of further research.

As an example of the correlation of error reconstruction values, the last two Snort [3] alerts listed below are very similar and the first is dissimilar from the last two. Since the last two alerts are numerically similar, they have very similar reconstruction errors (which are the same in this particular case: 3.1920 for both alerts, after

500 training epochs), but the last two have substantially different reconstruction errors from the first Snort alert below (which had reconstruction error 3.3388 in the same experiment).

```
[**] [1:716:10] TELNET access [**]
04/05-09:35:53.754855 172.16.112.50:23 -> 172.16.114.148:7298
TCP TTL:255 TOS:0x0 ID:28873 IpLen:20 DgmLen:55 DF
**AP*** Seq:0x367323BE Ack:0xF384C1BE Win:0x2238 TcpLen:20

[**] [1:1244:13] WEB-IIS ISAPI .idq attempt [**]
04/05-09:37:05.489493 172.16.117.103:8307 -> 208.160.10.123:80
TCP TTL:64 TOS:0x0 ID:4996 IpLen:20 DgmLen:549 DF
**AP*** Seq:0x336579CE Ack:0x7B33D444 Win:0x7D78 TcpLen:20

[**] [1:1245:10] WEB-IIS ISAPI .idq access [**]
04/05-09:37:05.489493 172.16.117.103:8307 -> 208.160.10.123:80
TCP TTL:64 TOS:0x0 ID:4996 IpLen:20 DgmLen:549 DF
**AP*** Seq:0x336579CE Ack:0x7B33D444 Win:0x7D78 TcpLen:20
```

From the list of reconstruction errors of a set of Snort alerts, we can use a simple algorithm to form discrete clusters of Snort events whose numerical representations are similar. This step is necessary because the autoassociator doesn't explicitly form clusters; there is a post-processing step required to do this. We discuss how this algorithm works in Section 3.3.

The parameters we used (and found to be optimal) were as follows:

- The number of epochs the autoassociator was trained for is 500,
- The number of hidden units we used was 8,
- We used the back propagation algorithm for training the autoassociator,
- We used 10,000 data items to train the autoassociator, to produce clusterings with more stable reconstruction error values, and
- The number of examples we were trying to cluster was 100, rather than 200 as in the April report – see Section C.4.

3.3 Clustering Autoassociator Output

The goal of the clustering algorithm is to input the reconstruction error values produced by the autoassociator, then turn the list of alerts into clusters with well-defined boundaries rather than a list real numbers with no obvious boundaries (but with intuitive clusters). To achieve this goal, we found that a heuristic approach worked well. We decided that any cohesive gaps in the list of sorted reconstruction error values would indicate a boundary between clusters, so we used that idea.

We formed discrete clusters such that every cluster obeys this rule: When you sort the reconstruction errors of the data items in the cluster, the difference between the reconstruction errors of any pair of items adjacent in the sorted list can be at most 0.0025. We developed this heuristic from the data, and we found that it worked well for our purposes.

We experimented with the size of the barrier between clusters before settling on 0.0025. We found that this number is depedent on the number of instances the clustering algorithm is considering at once. If there are more than 200 instances in the test set, the cluster barrier, by default 0.0025, should be reduced. If there are fewer than 100 instances in the test set, the cluster barrier can be increased above 0.0025 for better-formed clusters.

4 Results Analysis

For the demonstration of how the system works in this final report, we have used the 1999 DARPA [2] IDS data set instead of the incidents.org [1] data set we used in the previous report because DARPA 1999 is a published and well-regarded data set. In general, we found the DARPA 1999 data set, as fed through the Snort intrusion detection system, produced a more homogeneous set of Snort alerts, and this is necessarily reflected in the following example.

4.1 Analysis of Example Clustering

The most common types of alerts seen when the DARPA 1999 IDS data set is fed through the Snort IDS are attempts at accessing potentially dangerous CGI programs through the web server. We also found that, though many of these alerts were related, this wasn't always reflected in the clusters formed using our system. We diagnosed the problem and found that the main source of this problem is that most of the alerts that Snort reported happen to have significantly different values for the `tcpSeqNum` and `tcpAckNum` features within what should sometimes be a cohesive cluster. Because of this problem, we considered removing the `tcpSeqNum` and `tcpAckNum` attributes from the attribute list for the DARPA data set, as we previously mentioned in Section 3.1.2. In the end, we chose to remove these attributes from the attribute list because using these attributes only hobbled the performance of our system.

We will talk about meaningful clusters we found in the output, but we do not offer a full analysis of all of the clusters produced here because such an analysis can be tedious for the reader and we already offered this to the reader in the April report. (See Section C.4 for a full analysis of the algorithm's output on the incidents.org data set.)

We use the output of one run of our system on the DARPA data set. We've attached the full output of the algorithm as Annex B. There were 40 discrete clusters formed by our system. Many of these clusters contain only one alert.

The first cluster we discuss is as follows:

```
Cluster 2:  (3.4179)
--------------------
[**][1:718:7] TELNET login incorrect [**]
04/05-09:38:26.936288 172.16.114.50:23 -> 172.16.114.168:10332
TCP TTL:64 TOS:0x10 ID:2014 IpLen:20 DgmLen:66 DF
**AP*** Seq:0x801D2FDC Ack:0x6EDB5C7F Win:0x7C00 TcpLen:20
```

```
[**][1:716:10] TELNET access [**]
04/05-09:38:24.823813 172.16.114.50:23 -> 172.16.114.168:10332
TCP TTL:64 TOS:0x10 ID:1986 IpLen:20 DgmLen:52 DF
**AP*** Seq:0x801D2F4A Ack:0x6EDB5C38 Win:0x7C00 TcpLen:20
```

As you can see, these two telnet protocol alerts are obviously related to each other (the source and destination ports match, and the packets are temporally close). Snort doesn't see these two alerts as related, since they are reported as being generated by different Snort rules. These are similar types of alerts and their packet signature signatures are similar, so they are clustered together. It is clear that someone tried to log in to the telnet service, got their password wrong once, then were able to successfully log in. Seeing these alerts correlated in this way, it is clear to the administrator that this alert is likely a false alarm, though this might not be as apparent if the analyst were to see the first alert by itself. (It is important to remember that `ipSrc`, `ipDest`, `tcpSeqNum`, `tcpAckNum` and time attributes are not considered by the autoassociator, so these parameters can vary freely. The autoassociator clusters these alerts together without the use of these parameters, so we can use their values independently verify that are clusters are correct.) A single variable correlation engine might correlate other alerts generated by these IP addresses together, when the logical clustering of alerts says that only these two alerts should be clustered together.

The next cluster we'll look at is cluster 35, which has an average reconstruction error of 3.8779. In this cluster, there 3 alerts which look like this:

```
Cluster 35:  (3.8779)
--------------------
[**][1:621:6] SCAN FIN [**]
04/05-09:43:32.062101 208.240.124.83:33361 -> 172.16.112.50:59
TCP TTL:50 TOS:0x0 ID:15348 IpLen:20 DgmLen:40
******F Seq:0x0 Ack:0x0 Win:0x400 TcpLen:20

[**][1:621:6] SCAN FIN [**]
04/05-09:43:10.237953 208.240.124.83:61454 -> 172.16.112.50:50
TCP TTL:48 TOS:0x0 ID:17293 IpLen:20 DgmLen:40
******F Seq:0x0 Ack:0x0 Win:0xC00 TcpLen:20

[**][1:621:6] SCAN FIN [**]
04/05-09:43:13.474823 208.240.124.83:43993 -> 172.16.112.50:72
TCP TTL:38 TOS:0x0 ID:16119 IpLen:20 DgmLen:40
******F Seq:0x0 Ack:0x0 Win:0x400 TcpLen:20
```

These 3 alerts are all related, clearly. The individual alerts might not alert the IDS analyst to the broader reconnaissance being perpetrated against his system, but by seeing these alerts clustered together, there will be no doubt in the administrator's mind that an attacker is scanning for listening TCP services through the firewall using the well-known FIN port scan. This scan is featured as one of the port-scanning methods available in the widely-distributed nmap software. The attacker is trying to find out which TCP ports have services listening on the target host so that he can probe the host further, and possibly execute an attack to exploit a vulnerability if an open and unpatched service is found.

As well, there is a stray alert that is apparently unrelated to the other three in this cluster:

```
[**] [1:882:4] WEB-CGI calendar access [**]
[Classification:Attempted Information Leak] [Priority:  2]
06/08-13:56:15.520118 172.16.116.201:18978 -> 207.115.134.24:80
TCP TTL:63 TOS:0x0 ID:14610 IpLen:20 DgmLen:276 DF
**AP*** Seq:0x543A46E Ack:0x7682F730 Win:0x7D78 TcpLen:20
```

The presence of this alert in this cluster is an example of the 40-to-1 mapping problem we discuss in Section 3.2.

The last cluster we'll look at clusters together more numerically similar items. Here is the cluster:

```
Cluster 40:  (4.5508)
--------------------
[**][113:1:1] (spp_frag2) Oversized fragment,
probable DoS [**]
04/05-08:39:50.136064 202.77.162.213 -> 172.16.112.50
ICMP TTL:253 TOS:0x0 ID:1234 IpLen:20 DgmLen:1500
Frag Offset:  0x1FCC Frag Size:  0x05C8
```

There are six other alerts identical to this one in the cluster. (The only parameter that varies between the alerts is the time attribute.) Clearly all of these alerts are related as well. Again, an administrator looking at one of these alerts individually might falsely conclude that the oversized IP fragment was generated by a one-time fault in the system, but when the administrator sees these alerts correlated, he will be more suspicious since none of the alert's attributes vary. Similarly, if the administrator concludes that these are in fact false alarms, he can discount the entire group of alerts at once, rather marking each of them individually.

4.2 Numerical Results

We counted the number of errors we made by counting the number of alerts which were correctly clustered versus the number of alerts which were incorrectly clustered in a sample of test data. There are other ways of counting errors – such as counting the number of completely correct clusters – but we felt this method was the most accurate guage of our performance. We performed the analysis on the incidents.org data set because the Snort output from that data set was more interesting and varied. We had 200 alerts in our sample of test data.

We differentiate between two types of error in this analysis because they are of different levels of importance. We call the first type of error *separation error*, which occurs when a group of alerts should belong to a larger group of related alerts, but isn't clustered with the larger group. For instance, if we have a cluster A of 10 nmap scan alerts and a separate cluster B of 3 nmap alerts that are clearly related to cluster A, then we count 3 separation errors because the 3 alerts in B should have been grouped with the 10 alerts in A. We consider this type of error less damaging because, if this error occurs, the intrusion analyst must look at more clusters, but he will not miss important alerts hidden in a larger group.

The second type of error we find is *clustering error*, which occurs when two unrelated alerts are clustered together. For instance, if we have a mixed cluster of 10 nmap scans and 5 Squid scans where the alerts of the two types in the cluster are obviously not related, we count 5 clustering errors. We consider this error type more damaging because it may cause an intrusion detection analyst to miss important information.

After running our system on 200 test data instances from the incidents.org data set, we analyzed the results and found that we had 55 errors, for a 72.5% accuracy rate. If we consider the individual types of errors, we had 26 clustering errors and 29 separation errors.

Finally, we have hypothesized that the reconstruction error for similar alerts will be numerically close together. This accuracy rate shows the result, but to buttress this point, we offer a graph of the reconstruction error values versus the type of alert as visual proof.

You can see clearly from Figure C.1 that the reconstruction error values visually form clusters of similar alert types.

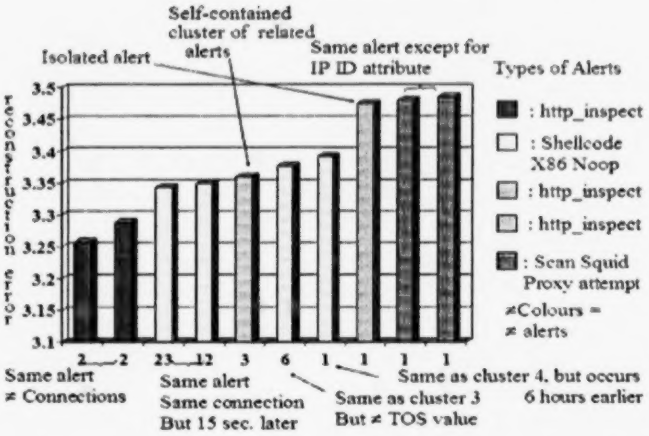


Figure 7: *Reconstruction Error Values*

4.3 Errors Made

One error, which we've already discussed in detail, is that we did not predict the 40-to-1 mapping problem. We simply diagnosed the problem after we recognized its presence. We could have performed an experiment to prove that this is in fact the problem with the classification. To perform this experiment, we would have analyzed how the reconstruction errors are determined in the cases where similar reconstruction error exists for dissimilar alerts. If the autoassociator output attributes contributing to the reconstruction error number were substantially different for the different alerts grouped together, we would have been able to conclude that the reconstruction error formula is the cause of the problem, and we could have considered a new mapping.

There are other ways to fix the 40-to-1 mapping as well. We could have used an established clustering algorithm on the autoassociator output (if this was permitted by the contract parameters), to see if the new clusters formed would be more cohesive.

We also considered a visualization algorithm which would visually decompose the reconstruction error number in 2 or 3 dimensions, but we didn't have time to flesh out this idea.

5 Conclusions and Future Work Recommendations

In conclusion, our system does work well for finding correlations between network events. The correlations that we find are stable, since the autoassociator is trained on so much data.

The performance of the system is good, but there is a problem where the 40-dimensional output of the autoassociator is mapped to the one-dimensional reconstruction error value range, causing a loss of information. This sometimes causes multiple types of alerts to be mapped to the same cluster when the alerts should form separate clusters. If we were able to devise a better system of clustering which would not require the 40-to-1 mapping, this problem would be solved. Clearly, this must be solved in future work.

Another subject of future work is how to update the autoassociator's internal representation without retraining the entire neural network. A nice aspect of neural networks is that they can store the essence of many data items without storing the actual items, and this advantage is lost if the data items must always be stored for retraining the neural network. There is active research into how to update neural networks with new data after training, and research into this area would be important to any system that might be deployed.

It is important to update the neural network as new data items arrive because if the neural network is initially trained on data from two years back, the results produced by the system will progressively worsen because the autoassociator won't recognize new data or new attack paradigms.

Another interest topic for future research would be to compare our system, using the autoassociator algorithm, to our system using a different clustering mechanism such as a support vector machine-based clustering algorithm. It would be interesting to explore options for machine learning-based correlation outside of the neural networks research area.

References

1. The incidents.org Data Set (Online). incidents.org. `http://www.incidents.org/logs/` (Mar. 14, 2004).

2. DARPA (1999). 1999 DARPA Intrusion Detection Evaluation Data Set Overview. *MIT: DARPA Intrusion Evaluation*.

3. Roesch, Martin (1999). Snort–Lightweight Intrusion Detection for Networks. In *Proceedings of LISA '99: 13th Systems Administration Conference*, Seattle, Washington: The USENIX Association.

4. The TCPDump Program (Online). tcpdump.org. `http://www.tcpdump.org/` (Jan. 20, 2004).

5. Stevens, W. Richard (1994). TCP/IP Illustrated : The Protocols, Vol. 1. Addison-Wesley.

6. Northcutt, Stephen (1999). Network Intrusion Detection: An Analyst's Handbook, New Riders Publishing.

7. Bejtlich, Richard (2000). Interpreting Network Traffic: A Network Intrusion Detector's Look at Suspicious Events. *first.org conference*.

8. Peter G. Neumann, Phillip A. Porras (1999). Experience with EMERALD to Date. In *1st USENIX Workshop on Intrusion Detection and Network Monitoring*, Santa Clara, California: SRI International Computer Science Lab.

9. Danyliw, Roman. ACID: Analysis Console for Intrusion Detections (Online). AIRCERT. `http://www.andrew.cmu.edu/user/rdanyliw/snort/snortacid.html` (Sept. 12, 2004).

10. SHADOW: Second Heuristic Analysis for Defensive Online Warfare (Online). NSWC. `http://www.nswc.navy.mil/ISSEC/CID/` (Sept. 12, 2004).

11. Giovanni Vigna, Richard A. Kemmerer (2001). NetSTAT: A Network-based Intrusion Detection System. *Reliable Software Group, Department of Computer Science, U.C. Santa Barbara*.

12. QuIDScor (Online). Qualys Inc.. `http://quidscor.sourceforge.net/` (Sept. 12, 2004).

13. Intellitactics NSM: Network Security Manager (Online). Intellitactics. `http://www.intellitactics.com/products/nsm_overview.html` (Sept. 12, 2004).

14. NFR: Network Flight Recorder (Online). NFR Security. `http://www.nfr.com/` (Sept. 12, 2004).

15. Wei Fan, Salvatore J. Stolfo, Wenke Lee and Miller, Matthew (2000). A Multiple Model Cost-Sensitive Approach for Intrusion Detection. *Department of Computer Science, Columbia University.*

16. Wenke Lee, Salvatore J. Stolfo (2000). A Framework for Constructing Features and Models for Intrusion Detection Systems. In *ACM Transactions on Information and System Security, Vol. 3, No. 4*, pp. 227–261. ACM.

17. Cohen, William (1995). Fast Effective Rule Induction. In *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 115–123.

18. Jim Georges, Anne H. Milley (1999). KDD'99 Competition: Knowledge Discovery Contest. *SAS Institute Inc.*

19. Eleazar Eskin, Michael Prerau; Leonid Portnoy, Andrew Arnold and Stolfo, Sal (2001). A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data. *Department of Computer Science, Columbia University.*

20. Fawcett, Tom (2003). ROC Graphs: Notes and Practical Considerations for Data Mining Researchers. *HP Laboratories.*

21. Peng Ning, Yun Cui (2002). An Intrusion Alert Correlator Based on Prerequisites of Intrusions. *Department of Computer Science, North Carolina State University.*

22. Herve Debar, Andreas Wespi (2001). Aggregation and Correlation of Intrusion-Detection Alerts. *France Telecom R&D, Zurich Research Laboratory.*

23. H. Debar, B. Feinstein, D. Curry (2004). The Intrusion Detection Message Exchange Format. *IETF Working Group.*

24. Alfonso Valdes, Keith Skinner (2001). Probabilistic Alert Correlation. In *RAID 2001, LNCS 2212*, pp. 54–68. Springer-Verlag.

25. Lippman, R.P. (1987). An Introduction to Computing with Neural Nets. In *IEEE ASSP Magazine*, pp. 4–22.

26. Demuth, H. and Beale, M. (2001). Neural Network Toolbox, MathWorks Version 4.

27. Zurada, J.M. (1992). Introduction to Artificial Neural Systems, New York: West Publishing Company.

28. Widrow, B. and Lehr, M.A. (1990). 30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation. In *IEEE Proceedings*, Vol. 78, pp. 1415–1442.

29. Martin T. Hagan, Mark Beale, Howard B. Demuth (1996). Neural Network Design, PWS Publishing Company.

30. Japkowicz, N (2001). Supervised Versus Unsupervised Binary-Learning by Feedforward Neural Networks, Ch. Forty-Two, pp. 97–122. Netherlands: Kluwer Academic Publishers.

31. The Neural Network Toolbox (Online). The MathWork, Inc.. `http://www.mathworks.com/products/neuralnet/` (Sept. 1, 2004).

32. Wall, Larry. The Perl programming language (Online). www.perl.com. `http://www.perl.com/` (Sept. 1, 2004).

33. The comp.ai.neural-nets FAQ (Online). comp.ai.neural-nets. `http://www.faqs.org/faqs/ai-faq/neural-nets/part2/` (Sept. 1, 2004).

34. Chih-Wei Hsu, Chih-Jen Lin, Chih-Chung Chang (2003). A Practical Guide to Support Vector Classification. *National Taiwan University*.

Annex A
Acronyms and Abbreviations

ACL	Access Control List
ANN	Artificial Neural Network
CGI	Common Gateway Interface
DARPA	Defense Advanced Research Projects Agency
DEFCON	Defence Conference
DDOS	Distributed Denial of Service
DOS	Denial of Service
DNS	Domain Name System
DRDOS	Distributed Reflected Denial of Service
DREnet	Defence Research Establishment Network
ECN	Explicit Congestion Control
FQDN	Fully Qualified Domain Name
IDS	Intrusion Detection System
IDWG	Intrusion Detection Working Group
ISN	The Initial Sequence Number
IP	Internet Protocol
HTML	Hyper-Text Markup Language
HTTP	Hyper-Text Transfer Protocol
HTRQ	Hypertext Request
MSE	Mean Square Error
OS	Operating System
ROC	Receiver Operating Characteristic
SVM	Support Vector Machine
TCP	Transmission Control Protocol
URL	Uniform Resource Locator
WWW	World Wide Web

Annex B
Full Output for DARPA Data

This appendix is the output of one run of our system on 100 Snort alerts generated from the DARPA [1] data set. This set of clusters is the set we used for the analysis in this final report.

```
Cluster 1:  (3.4135)
--------------------
[**][1:716:10] TELNET access [**]
04/05-09:23:27.772416 172.16.114.50:23 -> 172.16.114.169:21927
TCP TTL:64 TOS:0x10 ID:1157 IpLen:20 DgmLen:52 DF
**AP*** Seq:0x785D7AAE Ack:0xF9F08C1D Win:0x7C00 TcpLen:20

Cluster 2:  (3.4179)
--------------------
[**][1:718:7] TELNET login incorrect [**]
04/05-09:38:26.936288 172.16.114.50:23 -> 172.16.114.168:10332
TCP TTL:64 TOS:0x10 ID:2014 IpLen:20 DgmLen:66 DF
**AP*** Seq:0x801D2FDC Ack:0x6EDB5C7F Win:0x7C00 TcpLen:20

[**][1:716:10] TELNET access [**]
04/05-09:38:24.823813 172.16.114.50:23 -> 172.16.114.168:10332
TCP TTL:64 TOS:0x10 ID:1986 IpLen:20 DgmLen:52 DF
**AP*** Seq:0x801D2F4A Ack:0x6EDB5C38 Win:0x7C00 TcpLen:20

Cluster 3:  (3.4305)
--------------------
[**][1:716:10] TELNET access [**]
04/05-09:19:43.493159 135.8.60.182:23 -> 172.16.112.194:18513
TCP TTL:63 TOS:0x10 ID:20988 IpLen:20 DgmLen:55 DF
**AP*** Seq:0x8F655CBA Ack:0x2B6E96AC Win:0x7FC8 TcpLen:20

Cluster 4:  (3.4691)
--------------------
[**][1:716:10] TELNET access [**]
04/05-09:41:23.060326 196.227.33.189:23 -> 172.16.114.169:15185
TCP TTL:63 TOS:0x10 ID:35125 IpLen:20 DgmLen:55 DF
**AP*** Seq:0xC19C4B79 Ack:0x1E798307 Win:0x7FC8 TcpLen:20

Cluster 5:  (3.4949)
--------------------
[**][1:1244:13] WEB-IIS ISAPI .idq attempt [**]
```

```
04/05-09:37:05.489493 172.16.117.103:8307 -> 208.160.10.123:80
TCP TTL:64 TOS:0x0 ID:4996 IpLen:20 DgmLen:549 DF
**AP*** Seq:0x336579CE Ack:0x7B33D444 Win:0x7D78 TcpLen:20

[**][1:1245:10] WEB-IIS ISAPI .idq access [**]
04/05-09:37:05.489493 172.16.117.103:8307 -> 208.160.10.123:80
TCP TTL:64 TOS:0x0 ID:4996 IpLen:20 DgmLen:549 DF
**AP*** Seq:0x336579CE Ack:0x7B33D444 Win:0x7D78 TcpLen:20

Cluster 6:  (3.5024)
--------------------
[**][1:1201:7] ATTACK-RESPONSES 403 Forbidden [**]
04/05-09:03:34.081896 131.84.1.34:80 -> 172.16.115.5:32455
TCP TTL:63 TOS:0x0 ID:17057 IpLen:20 DgmLen:1001 DF
**AP*** Seq:0xE9B832EB Ack:0x65B63FA0 Win:0x7FE0 TcpLen:20

Cluster 7:  (3.5121)
--------------------
[**][1:1767:6] WEB-MISC search.dll access [**]
04/05-08:19:17.085991 172.16.114.168:2136 -> 208.221.32.46:80
TCP TTL:64 TOS:0x0 ID:7229 IpLen:20 DgmLen:444 DF
**AP*** Seq:0x65088253 Ack:0x9404D29F Win:0x7D78 TcpLen:20

[**][1:716:10] TELNET access [**]
04/05-08:44:32.680973 172.16.114.168:23 -> 202.77.162.213:1025
TCP TTL:64 TOS:0x10 ID:28842 IpLen:20 DgmLen:55 DF
**AP*** Seq:0x9FCA5A26 Ack:0xC26DE623 Win:0x7FC5 TcpLen:20

Cluster 8:  (3.5249)
--------------------
[**][1:718:7] TELNET login incorrect [**]
04/05-08:44:56.216972 172.16.114.168:23 -> 202.77.162.213:1025
TCP TTL:64 TOS:0x10 ID:30880 IpLen:20 DgmLen:66 DF
**AP*** Seq:0x9FCA5D30 Ack:0xC26DE6AB Win:0x7FE0 TcpLen:20

[**][119:13:1] (http_inspect) NON-RFC HTTP DELIMITER [**]
04/05-09:35:52.653184 172.16.117.111:7182 -> 206.132.25.51:80
TCP TTL:64 TOS:0x0 ID:2713 IpLen:20 DgmLen:378 DF
**AP*** Seq:0x5C7A74FB Ack:0xCBF37FF9 Win:0x7D78 TcpLen:20

[**][1:1201:7] ATTACK-RESPONSES 403 Forbidden [**]
04/05-08:37:11.631670 209.113.183.114:80 -> 172.16.114.168:12422
TCP TTL:63 TOS:0x0 ID:6889 IpLen:20 DgmLen:314 DF
**AP*** Seq:0xC394B652 Ack:0xF071380D Win:0x7FE0 TcpLen:20
```

```
[**][1:1201:7] ATTACK-RESPONSES 403 Forbidden [**]
04/05-09:08:06.574622 137.245.85.134:80 -> 172.16.114.148:4929
TCP TTL:63 TOS:0x0 ID:19138 IpLen:20 DgmLen:331 DF
**AP*** Seq:0x98C32BF4 Ack:0xFCEF4B9B Win:0x7FE0 TcpLen:20

[**][1:895:7] WEB-CGI redirect access [**]
04/05-09:35:08.610841 172.16.115.87:6189 -> 207.46.176.50:80
TCP TTL:64 TOS:0x0 ID:1586 IpLen:20 DgmLen:357 DF
**AP*** Seq:0x9F05CA0 Ack:0x608BC8A1 Win:0x7D78 TcpLen:20

[**][1:1643:6] WEB-CGI db2www access [**]
04/05-09:40:55.057487 172.16.114.207:14440 -> 206.41.140.162:80
TCP TTL:64 TOS:0x0 ID:12536 IpLen:20 DgmLen:363 DF
**AP*** Seq:0x7609C51E Ack:0xC3278E07 Win:0x7D78 TcpLen:20

[**][1:716:10] TELNET access [**]
04/05-08:45:22.667027 172.16.114.168:23 -> 202.77.162.213:1026
TCP TTL:64 TOS:0x10 ID:32791 IpLen:20 DgmLen:55 DF
**AP*** Seq:0x28967C4E Ack:0x8859DC4E Win:0x7FC5 TcpLen:20

[**][1:1643:6] WEB-CGI db2www access [**]
04/05-09:41:19.879467 172.16.114.207:14995 -> 206.41.140.162:80
TCP TTL:64 TOS:0x0 ID:13262 IpLen:20 DgmLen:360 DF
**AP*** Seq:0xA450FEC Ack:0xCFC4E095 Win:0x7D78 TcpLen:20

[**][1:1643:6] WEB-CGI db2www access [**]
04/05-09:40:40.977216 172.16.114.207:14359 -> 206.41.140.162:80
TCP TTL:64 TOS:0x0 ID:12172 IpLen:20 DgmLen:350 DF
**AP*** Seq:0xBC0CC18E Ack:0xA7190818 Win:0x7D78 TcpLen:20

[**][1:1201:7] ATTACK-RESPONSES 403 Forbidden [**]
04/05-08:20:38.103004 137.245.85.134:80 -> 172.16.114.168:2741
TCP TTL:63 TOS:0x0 ID:1523 IpLen:20 DgmLen:331 DF
**AP*** Seq:0x4D7B6355 Ack:0x8464CCD0 Win:0x7FE0 TcpLen:20

[**][1:1201:7] ATTACK-RESPONSES 403 Forbidden [**]
04/05-09:16:47.876431 137.245.85.134:80 -> 172.16.117.103:15709
TCP TTL:63 TOS:0x0 ID:23978 IpLen:20 DgmLen:331 DF
**AP*** Seq:0xB85A8E32 Ack:0x97893D5E Win:0x7FE0 TcpLen:20

[**][1:1201:7] ATTACK-RESPONSES 403 Forbidden [**]
04/05-08:17:53.370495 137.245.85.134:80 -> 172.16.117.103:1637
TCP TTL:63 TOS:0x0 ID:653 IpLen:20 DgmLen:331 DF
**AP*** Seq:0xFA59E98E Ack:0x8ABE5859 Win:0x7FE0 TcpLen:20
```

```
[**][1:1201:7] ATTACK-RESPONSES 403 Forbidden [**]
04/05-08:18:04.202755 137.245.85.134:80 -> 172.16.117.52:1643
TCP TTL:63 TOS:0x0 ID:691 IpLen:20 DgmLen:331 DF
**AP*** Seq:0xAAE53D8A Ack:0xFE6717F3 Win:0x7FE0 TcpLen:20

[**][1:1201:7] ATTACK-RESPONSES 403 Forbidden [**]
04/05-09:05:39.624242 209.113.183.114:80 -> 172.16.112.149:2181
TCP TTL:63 TOS:0x0 ID:17754 IpLen:20 DgmLen:314 DF
**AP*** Seq:0x1F1F86FF Ack:0xCE942BBC Win:0x7FE0 TcpLen:20

Cluster 9:  (3.5333)
--------------------
[**][1:1200:10] ATTACK-RESPONSES Invalid URL [**]
04/05-09:39:18.991664 207.200.75.201:80 -> 172.16.117.111:12370
TCP TTL:63 TOS:0x0 ID:37636 IpLen:20 DgmLen:468 DF
**AP*** Seq:0x57037225 Ack:0xDBE3C5D8 Win:0x7FE0 TcpLen:20

[**][1:716:10] TELNET access [**]
04/05-09:27:36.280723 172.16.112.194:23 -> 194.27.251.21:6734
TCP TTL:64 TOS:0x10 ID:44404 IpLen:20 DgmLen:55 DF
**AP*** Seq:0x881C4DCC Ack:0x677D8143 Win:0x7FC8 TcpLen:20

Cluster 10:  (3.5374)
--------------------
[**][1:1560:6] WEB-MISC /doc/ access [**]
04/05-09:07:05.576499 194.27.251.21:3398 -> 172.16.114.50:80
TCP TTL:63 TOS:0x0 ID:17866 IpLen:20 DgmLen:290 DF
**AP*** Seq:0xE964C568 Ack:0xD604FE9C Win:0x7D78 TcpLen:20

[**][1:1560:6] WEB-MISC /doc/ access [**]
04/05-09:07:05.542831 194.27.251.21:3334 -> 172.16.114.50:80
TCP TTL:63 TOS:0x0 ID:17852 IpLen:20 DgmLen:289 DF
**AP*** Seq:0x78E8B794 Ack:0xD69155CE Win:0x7D78 TcpLen:20

Cluster 11:  (3.5443)
--------------------
[**][1:2134:2] WEB-IIS register.asp access [**]
04/05-08:59:58.035368 172.16.116.44:29711 -> 207.46.131.142:80
TCP TTL:64 TOS:0x0 ID:2508 IpLen:20 DgmLen:340 DF
**AP*** Seq:0xBFA790A2 Ack:0xE598D64E Win:0x7D78 TcpLen:20

[**][1:972:8] WEB-IIS %2E-asp access [**]
04/05-08:59:58.035368 172.16.116.44:29711 -> 207.46.131.142:80
```

```
TCP TTL:64 TOS:0x0 ID:2508 IpLen:20 DgmLen:340 DF
**AP*** Seq:0xBFA790A2 Ack:0xE598D64E Win:0x7D78 TcpLen:20

[**][1:895:7] WEB-CGI redirect access [**]
04/05-09:35:46.834015 172.16.116.44:6934 -> 207.46.176.50:80
TCP TTL:64 TOS:0x0 ID:2405 IpLen:20 DgmLen:241 DF
**AP*** Seq:0x27377D3F Ack:0x3C69CC1E Win:0x7D78 TcpLen:20

[**][1:895:7] WEB-CGI redirect access [**]
04/05-08:27:44.084331 172.16.116.44:6201 -> 207.46.176.50:80
TCP TTL:64 TOS:0x0 ID:12271 IpLen:20 DgmLen:241 DF
**AP*** Seq:0xF160A1F1 Ack:0xD9A0FFED Win:0x7D78 TcpLen:20

[**][1:895:7] WEB-CGI redirect access [**]
04/05-08:59:41.641287 172.16.112.149:29506 -> 207.46.176.50:80
TCP TTL:64 TOS:0x0 ID:2164 IpLen:20 DgmLen:329 DF
**AP*** Seq:0x287444BE Ack:0xA53F9C73 Win:0x7D78 TcpLen:20

[**][1:972:8] WEB-IIS %2E-asp access [**]
04/05-09:38:52.044955 172.16.117.103:12165 -> 208.160.10.123:80
TCP TTL:64 TOS:0x0 ID:8598 IpLen:20 DgmLen:226 DF
**AP*** Seq:0xC01C0C18 Ack:0xCBF121C0 Win:0x7D78 TcpLen:20

[**][1:1200:10] ATTACK-RESPONSES Invalid URL [**]
04/05-09:42:41.552746 207.200.75.201:80 -> 172.16.117.111:16474
TCP TTL:63 TOS:0x0 ID:39683 IpLen:20 DgmLen:465 DF
**AP*** Seq:0x58C54B5F Ack:0xBC62D805 Win:0x7FE0 TcpLen:20

[**][1:895:7] WEB-CGI redirect access [**]
04/05-09:10:02.231330 172.16.114.169:7590 -> 207.46.176.50:80
TCP TTL:64 TOS:0x0 ID:20288 IpLen:20 DgmLen:258 DF
**AP*** Seq:0x98D169BD Ack:0x55378BA4 Win:0x7D78 TcpLen:20

[**][1:972:8] WEB-IIS %2E-asp access [**]
04/05-09:41:23.372473 172.16.117.103:15198 -> 208.160.10.123:80
TCP TTL:64 TOS:0x0 ID:13506 IpLen:20 DgmLen:226 DF
**AP*** Seq:0xD716C214 Ack:0xEACEACEF Win:0x7D78 TcpLen:20

[**][1:1560:6] WEB-MISC /doc/ access [**]
04/05-09:28:52.420068 135.8.60.182:7207 -> 172.16.112.100:80
TCP TTL:63 TOS:0x0 ID:25654 IpLen:20 DgmLen:301 DF
**AP*** Seq:0xC71E4B13 Ack:0x666B6C Win:0x7D78 TcpLen:20

[**][1:1560:6] WEB-MISC /doc/ access [**]
04/05-09:07:05.188198 194.27.251.21:3270 -> 172.16.114.50:80
```

```
TCP TTL:63 TOS:0x0 ID:17834 IpLen:20 DgmLen:225 DF
**AP*** Seq:0xC25AC485 Ack:0xCB89A60B Win:0x7D78 TcpLen:20

[**][1:1149:12] WEB-CGI count.cgi access [**]
04/05-08:42:08.429009 172.16.116.44:16530 -> 206.71.77.2:80
TCP TTL:64 TOS:0x0 ID:21088 IpLen:20 DgmLen:290 DF
**AP*** Seq:0x23AECA03 Ack:0x545F69C9 Win:0x7D78 TcpLen:20

[**][1:1560:6] WEB-MISC /doc/ access [**]
04/05-09:16:33.038674 196.227.33.189:4737 -> 172.16.114.50:80
TCP TTL:63 TOS:0x0 ID:20211 IpLen:20 DgmLen:227 DF
**AP*** Seq:0x512FCAA7 Ack:0x837141B7 Win:0x7D78 TcpLen:20

Cluster 12:  (3.5555)
--------------------
[**][1:895:7] WEB-CGI redirect access [**]
04/05-08:42:37.342305 172.16.116.44:17249 -> 209.186.73.3:80
TCP TTL:64 TOS:0x0 ID:21475 IpLen:20 DgmLen:262 DF
**AP*** Seq:0x6403101B Ack:0xF5837E01 Win:0x7D78 TcpLen:20

[**][1:895:7] WEB-CGI redirect access [**]
04/05-08:41:26.969925 172.16.116.44:16495 -> 209.83.166.88:80
TCP TTL:64 TOS:0x0 ID:20787 IpLen:20 DgmLen:241 DF
**AP*** Seq:0x2AA9CBE8 Ack:0x51617CE9 Win:0x7D78 TcpLen:20

Cluster 13:  (3.5615)
--------------------
[**][1:1213:5] WEB-MISC backup access [**]
04/05-09:31:00.244192 194.27.251.21:8243 -> 172.16.112.100:80
TCP TTL:63 TOS:0x0 ID:26848 IpLen:20 DgmLen:238 DF
**AP*** Seq:0xE3F4F26C Ack:0x685ED0 Win:0x7D78 TcpLen:20

[**][1:882:5] WEB-CGI calendar access [**]
04/05-09:16:42.236158 172.16.114.148:15465 -> 198.113.160.132:80
TCP TTL:64 TOS:0x0 ID:29554 IpLen:20 DgmLen:335 DF
**AP*** Seq:0x2E0B78A4 Ack:0xF779CB6E Win:0x7D78 TcpLen:20

[**][1:2134:2] WEB-IIS register.asp access [**]
04/05-09:17:22.256107 172.16.112.207:16090 -> 207.46.131.142:80
TCP TTL:64 TOS:0x0 ID:30394 IpLen:20 DgmLen:355 DF
**AP*** Seq:0x8EF79EA5 Ack:0x7E557E0A Win:0x7D78 TcpLen:20

[**][1:972:8] WEB-IIS %2E-asp access [**]
04/05-09:17:22.256107 172.16.112.207:16090 -> 207.46.131.142:80
```

```
TCP TTL:64 TOS:0x0 ID:30394 IpLen:20 DgmLen:355 DF
**AP*** Seq:0x8EF79EA5 Ack:0x7E557E0A Win:0x7D78 TcpLen:20

Cluster 14:  (3.5738)
--------------------
[**][1:882:5] WEB-CGI calendar access [**]
04/05-09:19:10.358703 172.16.114.148:17125 -> 198.113.160.132:80
TCP TTL:64 TOS:0x0 ID:32250 IpLen:20 DgmLen:335 DF
**AP*** Seq:0x7FFA53FA Ack:0x42D8A597 Win:0x7D78 TcpLen:20

Cluster 15:  (3.5773)
--------------------
[**][1:839:7] WEB-CGI finger access [**]
04/05-09:41:29.366953 135.8.60.182:9878 -> 172.16.112.100:80
TCP TTL:63 TOS:0x0 ID:35305 IpLen:20 DgmLen:294 DF
**AP*** Seq:0xBD664642 Ack:0x71F8C3 Win:0x7D78 TcpLen:20

Cluster 16:  (3.5874)
--------------------
[**][1:895:7] WEB-CGI redirect access [**]
04/05-09:20:42.054958 172.16.114.148:19202 -> 207.46.176.50:80
TCP TTL:64 TOS:0x0 ID:34218 IpLen:20 DgmLen:329 DF
**AP*** Seq:0xC90AE1DB Ack:0x8F943129 Win:0x7D78 TcpLen:20

Cluster 17:  (3.6146)
--------------------
[**][1:972:8] WEB-IIS %2E-asp access [**]
04/05-09:31:24.779533 172.16.112.207:1028 -> 207.46.145.24:80
TCP TTL:64 TOS:0x0 ID:52617 IpLen:20 DgmLen:470 DF
**AP*** Seq:0x2CD0E88 Ack:0xDEA59491 Win:0x7D78 TcpLen:20

Cluster 18:  (3.6222)
--------------------
[**][1:1226:4] X11 xopen [**]
04/05-08:45:54.318536 172.16.112.20:20696 -> 202.77.162.213:6000
TCP TTL:64 TOS:0x0 ID:35019 IpLen:20 DgmLen:52 DF
**AP*** Seq:0xDA852187 Ack:0x8591DDEB Win:0x7D78 TcpLen:20

[**][1:716:10] TELNET access [**]
04/05-09:25:45.355602 172.16.114.168:23 -> 206.48.44.50:4155
TCP TTL:64 TOS:0x0 ID:41721 IpLen:20 DgmLen:55 DF
**AP*** Seq:0x29964DEE Ack:0xC67B3E90 Win:0x7FC8 TcpLen:20
```

```
Cluster 19:  (3.6667)
--------------------
[**][119:13:1] (http_inspect) NON-RFC HTTP DELIMITER [**]
04/05-09:34:01.099161 172.16.115.5:4194 -> 199.95.209.99:80
TCP TTL:64 TOS:0x0 ID:57933 IpLen:20 DgmLen:379 DF
**AP*** Seq:0x7F62187F Ack:0x74517A6A Win:0x7D78 TcpLen:20

Cluster 20:  (3.7038)
--------------------
[**][1:718:7] TELNET login incorrect [**]
04/05-08:48:44.758564 172.16.112.50:23 -> 172.16.114.169:21604
TCP TTL:255 TOS:0x0 ID:11525 IpLen:20 DgmLen:57 DF
**AP*** Seq:0x211C02D1 Ack:0xDC59CEBD Win:0x2238 TcpLen:20

[**][1:716:10] TELNET access [**]
04/05-08:48:38.577677 172.16.112.50:23 -> 172.16.114.169:21604
TCP TTL:255 TOS:0x0 ID:11501 IpLen:20 DgmLen:55 DF
**AP*** Seq:0x211C024C Ack:0xDC59CE86 Win:0x2238 TcpLen:20

Cluster 21:  (3.7115)
--------------------
[**][1:716:10] TELNET access [**]
04/05-09:26:58.252941 172.16.112.50:23 -> 172.16.113.204:28740
TCP TTL:255 TOS:0x0 ID:5494 IpLen:20 DgmLen:55 DF
**AP*** Seq:0x3269F303 Ack:0x79771CAA Win:0x2238 TcpLen:20

Cluster 22:  (3.7148)
--------------------
[**][1:716:10] TELNET access [**]
04/05-09:00:33.114920 172.16.112.50:23 -> 172.16.113.204:29737
TCP TTL:255 TOS:0x0 ID:5205 IpLen:20 DgmLen:55 DF
**AP*** Seq:0x268921E2 Ack:0x3F5A1A31 Win:0x2238 TcpLen:20

Cluster 23:  (3.7216)
--------------------
[**][1:716:10] TELNET access [**]
04/05-09:01:39.463276 172.16.112.50:23 -> 172.16.112.207:31278
TCP TTL:255 TOS:0x0 ID:6019 IpLen:20 DgmLen:55 DF
**AP*** Seq:0x270AAB01 Ack:0xDDC14768 Win:0x2238 TcpLen:20

Cluster 24:  (3.7535)
--------------------
[**][1:716:10] TELNET access [**]
04/05-08:24:43.822154 172.16.112.50:23 -> 196.227.33.189:1877
```

```
TCP TTL:255 TOS:0x0 ID:18303 IpLen:20 DgmLen:55 DF
**AP*** Seq:0x16595518 Ack:0xD7465458 Win:0x2238 TcpLen:20

Cluster 25:  (3.7716)
--------------------
[**][1:716:10] TELNET access [**]
04/05-09:35:53.754855 172.16.112.50:23 -> 172.16.114.148:7298
TCP TTL:255 TOS:0x0 ID:28873 IpLen:20 DgmLen:55 DF
**AP*** Seq:0x367323BE Ack:0xF384C1BE Win:0x2238 TcpLen:20

Cluster 26:  (3.7760)
--------------------
[**][1:716:10] TELNET access [**]
04/05-08:54:26.729687 172.16.112.50:23 -> 172.16.113.105:25378
TCP TTL:255 TOS:0x0 ID:32001 IpLen:20 DgmLen:55 DF
**AP*** Seq:0x23C78764 Ack:0x140F9C63 Win:0x2238 TcpLen:20

Cluster 27:  (3.7970)
--------------------
[**][1:895:7] WEB-CGI redirect access [**]
04/05-08:56:56.367746 172.16.112.149:27533 -> 207.46.176.50:80
TCP TTL:64 TOS:0x0 ID:62615 IpLen:20 DgmLen:329 DF
**AP*** Seq:0xF23170EB Ack:0x1C36199C Win:0x7D78 TcpLen:20

[**][1:716:10] TELNET access [**]
04/05-09:33:46.101609 172.16.112.50:23 -> 172.16.113.84:4180
TCP TTL:255 TOS:0x0 ID:32285 IpLen:20 DgmLen:55 DF
**AP*** Seq:0x357BAF7A Ack:0x4D950A7 Win:0x2238 TcpLen:20

Cluster 28:  (3.8149)
--------------------
[**][1:621:6] SCAN FIN [**]
04/05-09:43:40.256971 208.240.124.83:37694 -> 172.16.112.50:23
TCP TTL:53 TOS:0x0 ID:44416 IpLen:20 DgmLen:40
******F Seq:0x0 Ack:0x0 Win:0x1000 TcpLen:20

Cluster 29:  (3.8243)
--------------------
[**][1:621:6] SCAN FIN [**]
04/05-09:43:30.971091 208.240.124.83:48284 -> 172.16.112.50:63
TCP TTL:57 TOS:0x0 ID:28262 IpLen:20 DgmLen:40
******F Seq:0x0 Ack:0x0 Win:0x1000 TcpLen:20
```

```
[**][1:621:6] SCAN FIN [**]
04/05-09:43:34.243272 208.240.124.83:37693 -> 172.16.112.50:23
TCP TTL:53 TOS:0x0 ID:61815 IpLen:20 DgmLen:40
******F Seq:0x0 Ack:0x0 Win:0x1000 TcpLen:20

[**][1:621:6] SCAN FIN [**]
04/05-09:43:48.487992 208.240.124.83:38051 -> 172.16.112.50:25
TCP TTL:52 TOS:0x0 ID:35754 IpLen:20 DgmLen:40
******F Seq:0x0 Ack:0x0 Win:0xC00 TcpLen:20

[**][1:895:7] WEB-CGI redirect access [**]
04/05-08:57:43.880520 172.16.116.44:28549 -> 143.166.224.44:80
TCP TTL:64 TOS:0x0 ID:64333 IpLen:20 DgmLen:256 DF
**AP*** Seq:0xFCB86D13 Ack:0xDC2292BB Win:0x7D78 TcpLen:20

[**][1:1560:6] WEB-MISC /doc/ access [**]
04/05-08:31:43.652665 172.16.112.50:32782 -> 172.16.112.100:80
TCP TTL:255 TOS:0x0 ID:26643 IpLen:20 DgmLen:328 DF
**AP*** Seq:0x1985D6AE Ack:0x32176D Win:0x2238 TcpLen:20

Cluster 30:  (3.8328)
--------------------
[**][1:621:6] SCAN FIN [**]
04/05-09:43:33.153367 208.240.124.83:57263 -> 172.16.112.50:8
TCP TTL:56 TOS:0x0 ID:35851 IpLen:20 DgmLen:40
******F Seq:0x0 Ack:0x0 Win:0xC00 TcpLen:20

[**][1:621:6] SCAN FIN [**]
04/05-09:43:14.561239 208.240.124.83:36030 -> 172.16.112.50:76
TCP TTL:51 TOS:0x0 ID:60556 IpLen:20 DgmLen:40
******F Seq:0x0 Ack:0x0 Win:0x800 TcpLen:20

[**][1:621:6] SCAN FIN [**]
04/05-09:43:09.166662 208.240.124.83:51945 -> 172.16.112.50:51
TCP TTL:37 TOS:0x0 ID:36064 IpLen:20 DgmLen:40
******F Seq:0x0 Ack:0x0 Win:0x1000 TcpLen:20

[**][1:621:6] SCAN FIN [**]
04/05-09:43:11.309436 208.240.124.83:40546 -> 172.16.112.50:45
TCP TTL:56 TOS:0x0 ID:24465 IpLen:20 DgmLen:40
******F Seq:0x0 Ack:0x0 Win:0xC00 TcpLen:20

Cluster 31:  (3.8424)
--------------------
```

```
[**][1:621:6] SCAN FIN [**]
04/05-09:43:21.650746 208.240.124.83:57624 -> 172.16.112.50:22
TCP TTL:36 TOS:0x0 ID:55673 IpLen:20 DgmLen:40
******F Seq:0x0 Ack:0x0 Win:0xC00 TcpLen:20

[**][1:621:6] SCAN FIN [**]
04/05-09:44:01.624466 208.240.124.83:60486 -> 172.16.112.50:83
TCP TTL:43 TOS:0x0 ID:56682 IpLen:20 DgmLen:40
******F Seq:0x0 Ack:0x0 Win:0x800 TcpLen:20

[**][1:621:6] SCAN FIN [**]
04/05-09:43:15.650830 208.240.124.83:57623 -> 172.16.112.50:22
TCP TTL:36 TOS:0x0 ID:60075 IpLen:20 DgmLen:40
******F Seq:0x0 Ack:0x0 Win:0xC00 TcpLen:20

[**][1:621:6] SCAN FIN [**]
04/05-09:43:28.789952 208.240.124.83:38097 -> 172.16.112.50:4
TCP TTL:42 TOS:0x0 ID:43310 IpLen:20 DgmLen:40
******F Seq:0x0 Ack:0x0 Win:0x400 TcpLen:20

[**][1:621:6] SCAN FIN [**]
04/05-09:43:29.881371 208.240.124.83:41499 -> 172.16.112.50:15
TCP TTL:35 TOS:0x0 ID:38312 IpLen:20 DgmLen:40
******F Seq:0x0 Ack:0x0 Win:0x800 TcpLen:20

[**][1:621:6] SCAN FIN [**]
04/05-09:43:54.504036 208.240.124.83:38052 -> 172.16.112.50:25
TCP TTL:52 TOS:0x0 ID:20076 IpLen:20 DgmLen:40
******F Seq:0x0 Ack:0x0 Win:0xC00 TcpLen:20

Cluster 32:  (3.8538)
--------------------
[**][1:621:6] SCAN FIN [**]
04/05-09:43:12.389758 208.240.124.83:56753 -> 172.16.112.50:71
TCP TTL:47 TOS:0x0 ID:26761 IpLen:20 DgmLen:40
******F Seq:0x0 Ack:0x0 Win:0x800 TcpLen:20

[**][1:621:6] SCAN FIN [**]
04/05-09:44:07.127593 208.240.124.83:63556 -> 172.16.112.50:44
TCP TTL:40 TOS:0x0 ID:37361 IpLen:20 DgmLen:40
******F Seq:0x0 Ack:0x0 Win:0xC00 TcpLen:20

Cluster 33:  (3.8619)
--------------------
```

```
[**][1:621:6] SCAN FIN [**]
04/05-09:44:06.037082 208.240.124.83:45920 -> 172.16.112.50:33
TCP TTL:43 TOS:0x0 ID:18886 IpLen:20 DgmLen:40
******F Seq:0x0 Ack:0x0 Win:0x800 TcpLen:20

Cluster 34:  (3.8678)
--------------------
[**][1:621:6] SCAN FIN [**]
04/05-09:44:04.945637 208.240.124.83:36225 -> 172.16.112.50:70
TCP TTL:45 TOS:0x0 ID:9103 IpLen:20 DgmLen:40
******F Seq:0x0 Ack:0x0 Win:0x1000 TcpLen:20

[**][1:621:6] SCAN FIN [**]
04/05-09:43:08.073616 208.240.124.83:43170 -> 172.16.112.50:3
TCP TTL:55 TOS:0x0 ID:11515 IpLen:20 DgmLen:40
******F Seq:0x0 Ack:0x0 Win:0x800 TcpLen:20

Cluster 35:  (3.8779)
--------------------
[**][1:621:6] SCAN FIN [**]
04/05-09:43:32.062101 208.240.124.83:33361 -> 172.16.112.50:59
TCP TTL:50 TOS:0x0 ID:15348 IpLen:20 DgmLen:40
******F Seq:0x0 Ack:0x0 Win:0x400 TcpLen:20

[**][1:1560:6] WEB-MISC /doc/ access [**]
04/05-09:01:50.499081 172.16.112.50:32792 -> 172.16.112.100:80
TCP TTL:255 TOS:0x0 ID:37016 IpLen:20 DgmLen:338 DF
**AP*** Seq:0x27228CEE Ack:0x4DAAB1 Win:0x2238 TcpLen:20

[**][1:621:6] SCAN FIN [**]
04/05-09:43:10.237953 208.240.124.83:61454 -> 172.16.112.50:50
TCP TTL:48 TOS:0x0 ID:17293 IpLen:20 DgmLen:40
******F Seq:0x0 Ack:0x0 Win:0xC00 TcpLen:20

[**][1:621:6] SCAN FIN [**]
04/05-09:43:13.474823 208.240.124.83:43993 -> 172.16.112.50:72
TCP TTL:38 TOS:0x0 ID:16119 IpLen:20 DgmLen:40
******F Seq:0x0 Ack:0x0 Win:0x400 TcpLen:20

Cluster 36:  (3.8907)
--------------------
[**][1:621:6] SCAN FIN [**]
04/05-09:44:02.763592 208.240.124.83:33711 -> 172.16.112.50:39
```

```
TCP TTL:38 TOS:0x0 ID:13313 IpLen:20 DgmLen:40
******F Seq:0x0 Ack:0x0 Win:0x400 TcpLen:20

Cluster 37:  (3.9022)
--------------------
[**][1:621:6] SCAN FIN [**]
04/05-09:44:03.855220 208.240.124.83:44454 -> 172.16.112.50:97
TCP TTL:43 TOS:0x0 ID:3024 IpLen:20 DgmLen:40
******F Seq:0x0 Ack:0x0 Win:0x800 TcpLen:20

Cluster 38:  (3.9176)
--------------------
[**][1:621:6] SCAN FIN [**]
04/05-09:43:47.392257 208.240.124.83:57734 -> 172.16.112.50:88
TCP TTL:52 TOS:0x0 ID:536 IpLen:20 DgmLen:40
******F Seq:0x0 Ack:0x0 Win:0xC00 TcpLen:20

Cluster 39:  (3.9786)
--------------------
[**][1:716:10] TELNET access [**]
04/05-09:38:40.989839 172.16.112.50:23 -> 194.7.248.153:8911
TCP TTL:255 TOS:0x0 ID:65031 IpLen:20 DgmLen:55 DF
**AP*** Seq:0x37B4BAF2 Ack:0x97D85EFD Win:0x2238 TcpLen:20

Cluster 40:  (4.5508)
--------------------
[**][113:1:1] (spp_frag2) Oversized fragment,
probable DoS [**]
04/05-08:39:50.136064 202.77.162.213 -> 172.16.112.50
ICMP TTL:253 TOS:0x0 ID:1234 IpLen:20 DgmLen:1500
Frag Offset:  0x1FCC Frag Size:  0x05C8

[**][113:1:1] (spp_frag2) Oversized fragment,
probable DoS [**]
04/05-08:39:50.191516 202.77.162.213 -> 172.16.112.50
ICMP TTL:253 TOS:0x0 ID:1234 IpLen:20 DgmLen:1500
Frag Offset:  0x1FCC Frag Size:  0x05C8

[**][113:1:1] (spp_frag2) Oversized fragment,
probable DoS [**]
04/05-08:39:50.246971 202.77.162.213 -> 172.16.112.50
ICMP TTL:253 TOS:0x0 ID:1234 IpLen:20 DgmLen:1500
Frag Offset:  0x1FCC Frag Size:  0x05C8
```

```
[**][113:1:1] (spp_frag2) Oversized fragment,
probable DoS [**]
04/05-08:39:50.287628 202.77.162.213 -> 172.16.112.50
ICMP TTL:253 TOS:0x0 ID:1234 IpLen:20 DgmLen:1500
Frag Offset:  0x1FCC Frag Size:  0x05C8

[**][113:1:1] (spp_frag2) Oversized fragment,
probable DoS [**]
04/05-08:39:50.308569 202.77.162.213 -> 172.16.112.50
ICMP TTL:253 TOS:0x0 ID:1234 IpLen:20 DgmLen:1500
Frag Offset:  0x1FCC Frag Size:  0x05C8

[**][113:1:1] (spp_frag2) Oversized fragment,
probable DoS [**]
04/05-08:48:30.809265 202.77.162.213 -> 172.16.114.50
ICMP TTL:253 TOS:0x0 ID:1234 IpLen:20 DgmLen:1500
Frag Offset:  0x1FCC Frag Size:  0x05C8

[**][113:1:1] (spp_frag2) Oversized fragment,
probable DoS [**]
04/05-08:48:30.864704 202.77.162.213 -> 172.16.114.50
ICMP TTL:253 TOS:0x0 ID:1234 IpLen:20 DgmLen:1500
Frag Offset:  0x1FCC Frag Size:  0x05C8
```

Annex C
Progress Reports

C.1 January Report

Neural Network Application to Network Event Correlation

Project Implementation Plan

C.1.1 Problem Overview

Network intrusion has become quite common in recent years and costs organizations millions of dollars each year. It is, therefore, of paramount importance to detect and prevent system probing or penetration attempts.

It is well known that false positives (known as "false alarms", intuitively) are very common in intrusion detection systems (IDSs). False positives are often great in proportion to true positives (alarms signifying real intrusions) and thus pose a very practical threat to the viability of IDSs by diminishing their usefulness. A security analyst who must sort through many false positives to find actual network intrusion attempts will have a reduced effectiveness at diagnosing real security risks. Further to this time lost separating the wheat from the chaff, the security analyst is less effective because she might miss an important intrusion attempt in her sorting effort.

One way to highlight the important intrusion attempts is to perform correlation analysis on the IDS log data. If two IDS events can be correlated with some level of certainty, it gives weight to the hypothesis that the correlated events are of interest. As well, if correlated events can be grouped together with some certainty such that the grouping has semantic value, it reduces the amount of work required by the security analyst by allowing her to analyse a group of events together.

Another benefit of correlation analysis is that it can improve the quality of the IDS. Using correlation analysis, events such as a slow SYN scan, where the attacker probes one port of a host every 10 minutes with a well-constructed TCP SYN packet, or distributed attacks, where the attacker has control of a number of IP addresses and uses them all in his attack, might become easier to detect. Conventional methods of intrusion detection and analysis might miss these stealthy attacks because of the problem of false positives and because of the apparent independence of the events generated by the IDS.

This study will focus on correlation of events generated by a given IDS. We will evaluate the applicability of neural networks to this task. In particular, we will attempt to use supervised and unsupervised neural network techniques to decide whether a particular event can be correlated to another event with a reasonable degree of certainty. We will compare the performance of our approaches to that of a previously designed correlation systems such as ACID, Shadow, netForensics, or Intellitactics NSM (in order to evaluate how promising neural networks are, both in terms of performance and ease of development). (Neural network event correlation systems are easier to design than knowledge-based ones since they perform the analysis of previous attacks themselves, rather than relying on analyses by human experts. As well, they can be more easily updated when new events take place than correlations done by their human-based counterparts.) We will also try using other, non-connectionist based, inductive systems in order to establish whether neural networks are, indeed, the best inductive tools to use for this problem.

Depending on the data available to us and the amount of time it requires to complete this first task satisfactorily, this study may include other efforts. In particular, we would like to study how to combine, or make homogenous, the outputs of different network devices (such as firewalls, host-based (rather than network-based) IDSs, or routers) to correlate their security warnings with those of the network IDS. As well, we'd like to consider how to effectively correlate events across different networks (produced by separate instances of the IDS).

C.1.2 Data

The data available from *http://incidents.org/logs/raw* will be used in our experiments. This data is raw, Snort-generated data, representing flagged network activity between April 2002 and October 2002. This data will be preprocessed using modified *Perl* scripts from
http://www.giac.org/practical/Mike_Bell_GCIA.doc and
http://www.giac.org/practical/chris_kuethe_gcia.html
to format the data into a form we can give to neural networks. We will try to represent all meaningful aspects of the Snort alert data in features to be given to the neural network. For example, given a Snort alert which reports that an IP packet containing a TCP header with both the SYN and FIN bits set, we would create a data item which contains a 1 for the SYN and FIN features.

As well, we will use data from the DARPA 1998 intrusion detection evaluation dataset to build the profile of a normal, non-Snort-alert-generating connection for our supervised learning experiments. To do this, we will extract all of the packets of a TCP connection using the TCP sequence number, IP source and destination addresses, and TCP source and destination ports of the TCP/IP packets.

C.1.3 Potential Areas of Investigation

The difficulty of the network event correlation problem makes it a rich research area to explore. In particular, there are various issues for us to consider:

1. Studying the feasibility of the proposed architecture
2. Establishing a Baseline
3. Data Distribution

If time permits:

1. Feature Space Distribution
2. Meta-Learning
3. Data Representation

Feasibility of the Architecture: The first aspect of our study will be to preprocess the data and assess the feasibility of the architecture proposed above based on the available data. It is quite possible that the architecture will have to be refined further once we are completely clear about the limitations of the data and the limitations of our representation of the data.

Establishing a Baseline: Once we settle on a given architecture for our overall system, we will question how the different components of this system will be implemented. Different choices are possible and we will test which of these choices would be most appropriate. Among the implementations we will consider are:

1. **Supervised Methods**: Multi-Layer Perceptrons, Decision Trees, Nave Bayes, Support Vector Machines (SVMs)
2. **Unsupervised Methods**: Auto-associators, Self–Organ-izing Maps (SOMs), Single–Class SVMs, Expectation–Maximization
3. **Supervised/Unsupervised Methods**: Radial Basis Function Networks (RBFs)

The evaluation of these various systems will be done by varying their different parameters and using various evaluation measures, possibly including accuracy, alarm rate, hit rate, and ROC analysis. (It may not be possible to test all these possibilities in the short amount of time given for this project. We will try the most likely fruitful ones as we go along.)

We will contrast the results obtained by the various components to those obtained using off-the-shelf network event correlation software such as: ACID, Shadow, netForensics, or Intellitactics NSM. This will give us an early estimate of how well neural networks or other inductive systems perform on network event correlation problems.

Data Distribution: Network intrusion data suffer from an acute class imbalance problem: there are many instances of false positives but only few instances of actual intrusions. Such a distributional bias affects many classifiers which tend to classify the large class well and the small class poorly. Several different approaches have been proposed to deal with the problem, including re-sampling approaches, cost-modifying approaches, and single-class (unsupervised) learning. We will attempt some of these approaches on our data. In general, our emphasis will be on the single-class learning strategy.

Feature Space Distribution: In a problem as complex as network intrusion correlation, it is wise to subdivide the problem as much as possible. We can easily do this from observable features. For example, we can learn several types of attacks separately. However, not all trends are readily observable, and yet subdividing the problem according to these unseen trends may prove useful. We will attempt to learn a subdivision of the feature space automatically and then apply specialized classifiers to each of these subdivided parts. This follows the previously proposed Mixture-of-Experts framework. However, that framework was proposed within a supervised context. Here, we could adapt it to the unsupervised case as follows: an autoassociator (or other unsupervised scheme), the gating unit, will be applied to the data. Typically, the autoassociator returns error reconstruction values that can be clustered into different sets. We will assume that this clustering represents the unobservable – though useful – subdivision of our feature space and we will apply a single specialized autoassociator, an expert, to each of these subdivisions. We expect such an approach to improve upon our recognition capability.

Meta-Learning: Meta-Learning consists of training a classifier to judge the outcome of running classification systems. It allows us to decide whether or not we should trust the decision issued by our classifiers. If sufficient data are available, we believe that this is a useful addition to any learning system and we will attempt to implement such a scheme.

Data Representation: In the main part of this project, we intend to conduct a lot of our work on raw processed data. However, it is clear that much could be gained from improving upon this given preliminary representation. This could be done using uninformed techniques (such as automated feature selection methods), but it could also be attempted using domain knowledge. For example, we could research a method to correlate the time element of different alerts in a meaningful way. This might involve making many important modifications to our proposed system.

This description represents only an initial take on the problem. It is possible, however, that, on the one hand, we did not assess the complexity of the problem well enough and that as a result, this research program can only be partially carried out. On the other hand, we should remember that prior to starting a project, certain issues may appear more realistic to study than they truly are. Conversely, new unforeseen issues may arise during the development of the research. This means that our description is only preliminary and is subject to change during the tenure of this project. We will, nonetheless, use it as a guide for our research and adhere to it whenever possible.

C.1.4 Outline Project Timeline

1. **December 29-January 29**: Preliminary reading; gathering of the data; assessment of the problem.
2. **January 29-February 29**: Data preparation; implementation of the overall system's architecture. Establishing results using off-the-shelf network event correlation software.
3. **February 29-March 29**: Implementation of the overall system's components using neural network architectures. Preliminary testing of the system.
4. **March 29-April 29**: Experiments varying the various parameters of each components as well as their architectures so as to settle on an optimal implementation.
5. **April 29-May 29**: Preliminary experiments with class imbalance corrections and changes of data representation; preparation of the final report and presentation.

We will choose one of these systems according to its availability, ease of use and reported performance. SRI's Emerald would have been a great tool to test as well, but unfortunately, it is not currently available for distribution.

C.2 February Report

Neural Network Application to Network Event Correlation

Progress Update

C.2.1 Overview of Report

After solidifying our view of how to approach the problem, we started work on processing the Snort-generated alert data into a format usable by a machine learning algorithm. We settled on a simple method of representing the alerts data in numerical form with the caveat that our way of processing data will evolve. We discuss our current methods of processing the alert data in Section C.2.2.

We then worked on establishing a baseline to which we could compare our results. We've considered many pieces of software, of which many were inappropriate for our task. In Section C.2.3 we elaborate on our findings and state limitations to our further work.

We've had progress in further defining our problem. There are many ways to correlate Snort-generated alerts. Existing pieces of software already correlate alerts (1) based on simple rules (such as, "cluster all alerts generated by a particular IP address"), (2) using external device data (such as correlating firewall warnings against IDS alerts), (3) using external network or database data (such as correlating IDS alerts against known product vulnerabilities, using a vulnerability database), (4) correlating different types of IDS alerts generated by one attacker, and (5) using statistical methods to discover new attack paradigms or new attack tools by correlating seemingly unrelated alerts.

Given all of these different interpretations of network event correlation, the clear trend common to all is that they all benefit the IDS analyst by allowing him to focus more on real attacks and by allowing him to better manage the risk associated these attacks. This goal of producing something very useful for the IDS analyst will be our motivation when considering any scheme.

Finally, in Section C.2.4 we discuss some preliminary analysis of the *incidents.org* dataset.

C.2.2 Data Processing

As discussed in our project plan, we are using the raw, unclassified Snort-generated alert data from *incidents.org* as our primary data source. This data is first read using

Snort, then output as textual alert data to *Perl* scripts that can understand alerts and can format the data for use with a machine learning algorithm.

To illustrate precisely how this transformation is made, and what information is encoded, we present an example. (The data items we output can be normalized (each feature linearly mapped to the interval [0,1]) – an important step in ensuring the numerical methods used in the machine learning classifier produce accurate results – but we present a non-normalized data item here for readability.)

The following *Snort*-generated alert was processed with our scripts to generate a row of numbers, where each number corresponds to a predetermined feature. The alert below was generated by Snort after a possible attacker probed TCP port 3128, likely in hope of finding an active service (specifically: the Squid web proxy daemon).

```
[**] [1:618:5] SCAN Squid Proxy attempt [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-16:00:56.996507 66.159.18.66:43518 -> 170.129.50.120:3128
TCP TTL:53 TOS:0x0 ID:50174 IpLen:20 DgmLen:60 DF
*****S* Seq:  0xBF3AC0C Ack:  0x0 Win:  0x16D0 TcpLen:  40
TCP Options (5) => MSS: 1460 SackOK TS: 48656370 0 NOP WS: 0
```

The *Perl* scripts we've created encode this alert information as predetermined features. We've created a table for all the features of the data item. As an example of how to read the table, in the data below, the number 2860593784 is the IP destination address ("ipDest") given in the alert. (The IP address is 170.129.50.120 before being encoded.)

Feature	Value	Feature	Value
snortGenId	1	snortSigId	618
snortSigRev	5	snortAlertClass	2
snortPriority	2	ipSrc	1117721154
portSrc	43518	ipDest	2860593784
portDest	3128	ipIsIcmpProtocol	0
ipIsIgmpProtocol	0	ipIsTcpProtocol	1
ipIsUdpProtocol	0	ipLen	20
ipDgmLen	60	ipId	50174
ipTos	0	ipTtl	53
ipOptLsrr	0	ipPacketDefrag	1
ipReserveBit	0	ipMiniFrag	0
ipFragOffset	0	ipFragSize	0
icmpCode	0	icmpId	0
icmpSeq	0	icmpType	0
tcpAckNum	0	tcpFlag1	0
tcpFlag2	0	tcpFlagUrg	0
tcpFlagAck	0	tcpFlagPsh	0
tcpFlagRst	0	tcpFlagSyn	1
tcpFlagFin	0	tcpLen	40
tcpSeqNum	200518668	tcpWinNum	5840
tcpUrgPtr	0	tcpOptMss	1460
tcpOptNopCount	0	tcpOptSackOk	1
tcpOptTs1	48656370	tcpOptTs2	0
tcpOptWs	0	tcpHeaderTrunc	0
udpLen	0		

As you can see, there are 49 features for each alert we encode. This number will likely change, as we determine more features to include (such as ones computed in non-obvious ways), or as we acknowledge that a particular feature impedes desired correlation.

There are too many features here to discuss each at length, so we will talk only about a few select features. We'll discuss the `ipIs*Protocol` set of features, the `tcpFlag*` set of features, and the `ip{Src,Dest}` features.

The features `ipIsIcmpProtocol`, `ipIsIgmpProtocol`, `ipIsTcpProtocol`, and `ipIsUdpProtocol` are telling binary features. If a data item has the value one for the `ipIsTcpProtocol` feature, this indicates that the Snort alert signifies an IP packet containing a TCP payload. Since the Squid proxy listens on a TCP port, this feature has value 1 for this data item. A Snort alert for a non-TCP IP packet will have `ipIsTcpProtocol` set to zero. Only one of the `ipIs*Protocol` features will be set for a given data item.

As you can see from the Snort alert, the TCP header in this packet has only the Syn flag set. (The string `******S*` indicates this.) This information is encoded in the `tcpFlag*` group of features. `tcpFlagSyn` is set to 1 and the rest of the flags are set to 0.

The `ipSrc` and `ipDest` features represent the IP source and destination addresses for a Snort alert. This information is encoded as follows: the IP address $a.b.c.d$ is encoded as $n = 256^3 a + 256^2 b + 256c + d$. (For example, the destination IP address in the data item above has this relationship: $2860593784 = 256^3 * 170 + 256^2 * 129 + 256 * 50 + 120$.) This has desired property that addresses within the same class C network will be closer together than addresses in different class C networks. Also, two addresses within a class C network will likely be closer together than two addresses in the greater class B network (which don't share the class C). This property can approximate distance between an attacker's targets, since an attacker will often target a particular network. But, as mentioned, if this feature turns out to hinder the clustering algorithm, it will not be considered.

C.2.3 Data Sets

In this task of alert correlation, the issue of adequate data recurs. In many data mining tasks, the problem of benchmarking how well a constructed system performs is solved by comparing the generated results against labelled testing data. Our primary data set, from *incidents.org*, does not allow us this luxury. The set of just over 30,000 raw alerts has not been processed (further than removing private information) or categorised in a way that could be used as a basis for comparison. As well, the alerts were generated on a network foreign to us, which can make the task of recognising meaningful correlation more difficult.

As we specified in the project proposal, we planned to use existing software to perform event correlation for us, and then compare our results to those of the existing software. After further investigating the software we originally proposed, we've discovered two of the packages (netForensics and Intellitactics NSM) are not readily available without paying for the software.

We've looked intensively at the ACID system, and decided that the correlation abilities of this software package are too primitive to act as a meaningful baseline. The extent of the correlation abilities of ACID are to display all alerts with a set IP address or to display all alerts of a particular type. There are other similar tasks the software can perform, but the results returned by these simple heuristic searches are not always useful or correct.

Another package we've considered using as a baseline for comparison is the freely-available QuIDScor package, which uses external data sources such as the Common

Vulnerability and Exposure (CVE) references to correlate alerts for more meaningful results. The QuIDScor system can also correlate alerts against a list of known network services to assess the importance of an alert. Although these novel methods of correlation would be very useful in determining the priority and the context of an alert, we do not plan to use similar data sources in our research, so we expect that our correlations will differ significantly in character from those produced by QuIDScor. As such, we feel that at this time QuIDScor will not be feasible as a baseline package.

The final package we're considering is the (also freely available) Shadow IDS system which is known to produce simple correlation like those found in ACID, but as well more advanced correlation such as using clever, statistically-based heuristics sometimes involving events over time. We think this package may be able to produce correlations that will be meaningful when comparing to those produced by our own software, but we will need to investigate this package further.

In short, to make our report rigourous, we must compare our devised methods against existing software packages. This comparison will almost certainly require us to act as human experts in judging the value of our correlations against the value of correlations produced by another package.

C.2.4 Initial Analysis

One of our first experiments involved getting to know the actual data better before finding correlations in it on a larger scale, and getting a feel for what kind of results we may be able to produce. For this experiment, we took the first 200 data items (of the 30,000+) and ran the *k*-means clustering algorithm on the data, setting the number of output clusters to five. (To perform this experiment and analyse the results, we used the freely-available WEKA data mining package.)

In the *k*-means clustering algorithm, you set the number of clusters you wish to produce, then the algorithm chooses *k* points randomly then assigns data items to each of the clusters by calculating which data items have the shortest Euclidean distance to the random points. The center of each cluster is updated to reflect the new cluster as more data points are assigned to the cluster. The algorithm is run a number of times to determine which data items perpetually end up in the same cluster. With this fairly simple clustering algorithm, we were able to find some trends in the data which we will be able to exploit with more complex, performance-oriented algorithms.

The five clusters distributed the data items as follows:

Cluster No.	Number of Data Items	Percentage of Total
0	17	9%
1	35	18%
2	42	21%
3	7	4%
4	99	50%

Each of the five clusters produced by *k*-means had clearly visible trends. To view this, look at a plot of the clusters (*y*-axis) formed against the `portSrc` feature (*x*-axis). We see that for cluster 2 and 3, the data is tightly bound to a small area. For cluster 4, the data is tightly bound to 2 small areas, and for the other two clusters, 0 and 1, we can still see that there are clots of data, even if the variance is greater.

The reason this data forms into tight clusters when examining the `portSrc` attribute is because many of the alerts in the 200 data items are alerts which relate to services on specific TCP ports. We see this by looking at representative items from each cluster.

An alert representative of both clusters 3 and 4 is the following:

```
[**] [1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:55:48.596507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:47256 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA0850378 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
[Xref=> http://www.whitehats.com/info/IDS181]
```

This alert occurred almost fifty times in the data set, and it seems that it is most likely caused by transmission of a file containing many 0x90 (NOP) bytes to a non-standard port. (The time signature of the alerts doesn't seem consistent with a normal or uniform distribution, but all the alerts happen in a 34-second window.) The *k*-means algorithm was correct in clustering these alerts together, but ideally it should have created only one cluster, rather than two, given the homogeneity of the alerts. (These alerts are almost entirely homogenous because they have many features constant among them, including the source and destination IP addresses and TCP ports.)

Our view is that correlating the data into clusters will be valuable for both true positives and false positives. The benefit of correlating true positives to one another is obvious, but the benefit of doing so to false positives is less obvious. We feel that if false positives can be correlated successfully, it will cut down the work of the analyst. Instead of sorting through many individual false positives, the analyst can simply look at the cluster as a whole to determine whether a set of alerts are false positives, saving his time for real attacks.

Cluster 0 (of size 17) consists of alerts similar to the following:

```
[**] [119:12:1] (http_inspect) APACHE WHITESPACE (TAB) [**]
11/15-05:01:20.766507 170.129.50.120:63175 -> 207.68.164.250:80
TCP TTL:125 TOS:0x0 ID:5602 IpLen:20 DgmLen:932 DF
**AP*** Seq:0xC9E02861 Ack:0x3015901B Win:0xFA61 TcpLen:20
```

There were multiple types of alerts in this category, but most of them dealt with sending non-standard input to an Apache server at port 80.

The clot of data in the middle of the row representing cluster 4 is dominated by data items which look like the following:

```
[**] [1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority:  3]
11/14-23:55:50.456507 255.255.255.255:31337 ->
170.129.161.133:515
TCP TTL:14 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq:  0x0 Ack:  0x0 Win:  0x0 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS203]
```

And the clot of data at the far left of the row representing cluster 4 has many data items which look like the following:

```
[**] [1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority:  2]
```

```
11/14-12:14:46.366507 210.66.117.5:80 -> 170.129.81.112:80
TCP TTL:47 TOS:0x0 ID:51968 IpLen:20 DgmLen:40
**A**** Seq: 0x400 Ack: 0x0 Win: 0x578 TcpLen: 20
[Xref=> http://www.whitehats.com/info/IDS28]
```

These different clots of data seem to be mostly unrelated, so separating them would have been valuable. This separation will probably happen as the value *k* in the *k*-means algorithm is increased. Clearly both of these alerts are valid, important alerts. Because we are working with such a small set of data in this experiment, it is difficult to tell exactly how *k*-means will perform with a larger dataset, and a sufficiently larger *k*. Each of the clusters produced so far represents (loosely) not only one attack method, but one attack. If different attacks using the same tool can be discerned in a larger dataset, we will be happy. It may be that after performing a high level clustering, we will find value in performing another clustering on one of the particular clusters produced by the first clustering.

The last cluster (cluster 1) contained many instances of two different alert types. The first is similar to the following:

```
[**] [1:523:4] BAD-TRAFFIC ip reserved bit set [**]
[Classification:Misc activity] [Priority: 3]
11/14-23:28:20.656507 200.200.200.1 -> 170.129.53.47
TCP TTL:241 TOS:0x0 ID:0 IpLen:20 DgmLen:40 RB
Frag Offset: 0x0864 Frag Size: 0x0014
```

And the second looked similar to this, or the Squid Proxy alert shown earlier inSection C.2.2:

```
[**] [1:615:5] SCAN SOCKS Proxy attempt [**]
[Classification:Attempted Information Leak] [Priority: 2]
11/14-16:00:56.996507 66.159.18.66:43521 -> 170.129.50.120:1080
TCP TTL:53 TOS:0x0 ID:2662 IpLen:20 DgmLen:60 DF
*****S* Seq: 0xBF46594 Ack: 0x0 Win: 0x16D0 TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 48656370 0 NOP WS: 0
[Xref=> http://help.undernet.org/proxyscan/]
```

This situation is probably similar to the situation seen with cluster 4, where two disparate types of alerts are grouped together because the *k* value we used was too small.

Finally, we see from these preliminary experiments focused on visualisation of the data that clustering may be effective in correlating Snort alerts, especially with a more sophisticated algorithm, such as one that can determine the number of cluster automatically.

C.3 March Report

Neural Network Application to Network Event Correlation

Progress Update

C.3.1 Overview of Report

In this report, we build on the ideas presented in our previous report to construct a new system with sufficient complexity to prevail over the difficulties inherent in this problem. After exploring the data further than what we presented in the last report, we came to understand that the simpler hypothetical systems we'd envisioned will not be sufficient to produce valuable results. The system we detail here uses the idea of clustering similar *Snort* events we presented in the last report as the first step in a process towards producing meaningful network event correlations.

In Section C.3.2 of this report, we present a system for producing dynamic, simple clusterings of events using an elegant piece of neural network architecture known as the autoassociator. The autoassociator produces robust results and works quite well for this task. In Section C.3.3, we explain how the autoassociator produces better results than those produced by the *k*-means algorithm used in the last report. In Section C.3.4, we give an example to illustrate the ideas presented in Section C.3.2 and Section C.3.3.

Machine learning algorithms analyze sets of numbers and infer that numerically similar data items are of the same class. Used on its own, the encoding for *Snort* events we presented in the last report tends to induce clusterings that are very sensitive to numerical dissimilarities, so that different types of *Snort* alerts almost always end up in different clusters. To produce more interesting correlations, we transform each of the clusters produced by the autoassociator into new individual data items, and then find correlation between these new data items using a different set of attributes that we'll construct. This basic idea is expanded in Section C.3.5, and implementation details for the scheme are addressed in Section C.3.6.

C.3.2 Clustering With The Autoassociator

The task of clustering numerically similar *Snort* events is key to our proposed system. Since our goal is to recognize correlation between groups of events, it is important to first determine cohesive groups. We will show how the autoassociator, an elegant neural network design, can be used for exactly this task.

The autoassociator is a piece of feedforward neural network architecture whose design is specialized for recognizing training input of one class rather than discriminating between training input of different classes. The autoassociator is known as an unsupervised machine learning algorithm because the classifier is trained on one class of data, so that it can recognize that class of data, then it's tested on more than one class of data. The autoassociator is able to compute the reconstruction error for a given data item, using the formula $\sum_{i=1}^{d}[x_i^{Test} - f_i(x^{Test})]^2$, where d is the number of attributes of the data items, x_i^{Test} is the value of the i^{th} attribute of data item x^{Test} and $f_i(x^{Test})$ is the value of the i^{th} output of the neural network with input x^{Test}. After training, the autoassociator is able to recognize the class represented in the training data because data items from the training class will have low reconstruction errors compared to data items from unseen classes. The theory holds that data items from classes without representative training examples will have higher reconstruction errors. A good treatment of using the autoassociator for this type of classification is "Supervised Versus Unsupervised Binary-Learning by Feedforward Neural Networks" by N. Japkowicz (2001) from the journal Machine Learning 42, pages 97-122.

In our task, we will use the autoassociator for clustering, rather than for binary classification. The difference in reconstruction error of two given data is positively correlated to the general numerical difference of those data. In our task, if the numerical representation of two *Snort* events (see Section C.2.2, Data Processing of our last report) is similar, the autoassociator will produce similar reconstruction errors. The converse is also true: numerically dissimilar *Snort* events will produce dissimilar reconstruction errors.

For example, the first two *Snort* events listed below, being very similar, and thus numerically similar, have very similar reconstruction errors (4.2247 and 4.2244 respectively after 500 training epochs with much variance among the training data items), but the first two have substantially different reconstruction errors from the third *Snort* event below (4.1465 in the same experiment).

```
[**] [1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-10:10:18.856507 61.218.161.210:80 -> 170.129.19.170:80
TCP TTL:48 TOS:0x0 ID:30943 IpLen:20 DgmLen:40
**A**** Seq:  0x278 Ack:  0x0 Win:  0x578 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS28]

[**] [1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-10:10:23.856507 163.23.238.9:80 -> 170.129.19.170:80
TCP TTL:44 TOS:0x0 ID:31290 IpLen:20 DgmLen:40
**A**** Seq:  0x300 Ack:  0x0 Win:  0x578 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS28]
```

```
[**] [1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority:  3]
11/14-10:10:22.596507 255.255.255.255:31337 ->
170.129.195.178:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq:  0x0 Ack:  0x0 Win:  0x0 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS203]
```

From a list of reconstruction errors for a set of *Snort* events, we can use a simple algorithm to form discrete clusters of *Snort* events whose numerical representations are similar. (We discuss how to implement this algorithm in further detail as part of the example in Section C.3.4.) The goal of this algorithm is to cluster similar *Snort* events so that the IDS analyst can deal with these groups of events in a different way than they would a singular event. Clustering can also be done by hand quite easily, using a visual representation of the data, as you will see clearly from the simple graph in the next section.

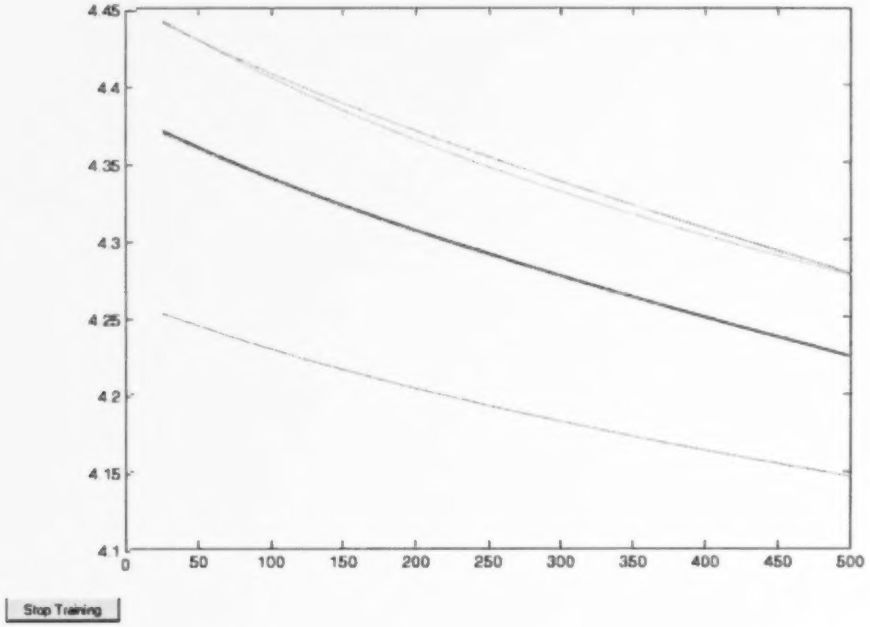
The reason this task of clustering is best done inductively by a machine learning algorithm, rather than deduced from attributes of the *Snort* event, is subtle. For instance, it might not be clear to the reader that we shouldn't simply form clusters based on the type of alert that *Snort* reports. It might seem like the clustering algorithm as presented will simply cluster all of the "SCAN nmap" alerts together in one cluster and all the "Invalid TCP flag combination" alerts in another cluster – since they are very similar numerically – but the clustering algorithm is sensitive enough to numerical differences between alerts of the same type that it will separate events from different connections into different clusters. As well, the algorithm is smart enough to cluster together alerts of different types which are very similar numerically. We present an example in Section C.3.4 to help show the truth in these assertions.

C.3.3 Comparison of Clusterers

In the Initial Analysis section of our last report, we used the simple *k*-means algorithm from the freely available *WEKA* package to explore the natural clusters that exist in the *incidents.org* data set.

Throughout our experiments, we've found that the autoassociator produces systematically better results than *k*-means, with fewer parameters to adjust. Namely, when using the autoassociator, you don't need to specify how many clusters to create, whereas with *k*-means this step is necessary.

In the graph below, produced by *Matlab*'s graphing facilities after obtaining results using tools from the *Neural Network Toolbox*, we can see there are three very clear classes of data, and it turns out that none of the ten data items are clustered incorrectly here. But experiments with the simple k-means algorithm showed that it consistently had an unacceptable level of noise (that is, misclustered items) in the *output*. k-means is convenient graphical method for quickly exploring data, but it isn't a stable method for producing statistically-strong results. (The y-axis in the graph below is the calculated reconstruction error for each item, and the x-axis is the number of epochs the neural network was trained for.)

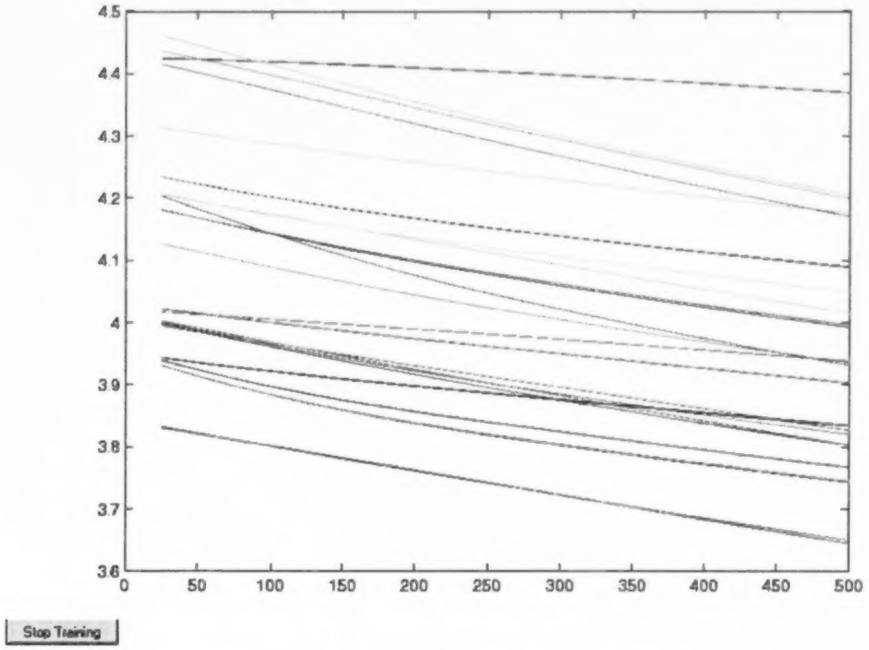


Although the autoassociator is very good at clustering numerically-like data items, the *Snort* event encoding we pass to the autoassociator makes it too sensitive for our eventual purpose, and splits related events into different clusters. It splits related events into different clusters because sometimes the encoding masks these valuable relationships by making the data numerically dissimilar. We will see an example of this sensitivity as part of the example in the next section, and we'll discuss how to solve this problem in Section C.3.5 and Section C.3.6.

C.3.4 Clustering Example

To illustrate better how the autoassociator forms clusters, and to show the types of relationships between the formed clusters, we present an example.

Our set of *Snort* alerts is a small subset of the *incidents.org* data; for the experiment, we uses 200 alerts from the dates November 9, 2002 to November 15, 2002. We train the autoassociator for 500 epochs on the first 100 alerts, and graph the test data at every 25-epoch interval. The set of test data is the last 100 alerts, which are alerts generated between the approximate times 22:00 on November 14 and 08:00 on November 15; so, the test data represents a *Snort* IDS over a 10 hour interval. See the graph below, where the x-axis is the epochs and the y-axis is the reconstruction error.



To form cohesive clusters within this set of *Snort* events, we used a heuristic. We formed discrete clusters using the rule: For every cluster, when you sort the reconstruction errors of the data items in the cluster, the difference between the reconstruction errors of any pair of items adjacent in the sorted list can be at most 0.0025. We developed this heuristic from the data, and we found that it worked very well, but we plan to experiment with it further – it would be best if this parameter were determined automatically, if possible.

From the reconstruction errors of the *Snort* events at epoch 500, we determined that the algorithm forms 23 clusters. For all but three of the clusters, there is only one type of *Snort* alert in the cluster; we note the three non-homogenous clusters below.

Cluster No.	No. Events	Representative Event
1	10	SHELLCODE x86 NOOP
2	4	SHELLCODE x86 NOOP
3	6	SHELLCODE x86 NOOP
4	12	SHELLCODE x86 inc ebx NOOP
5	1	SHELLCODE x86 inc ebx NOOP
6	11	BAD-TRAFFIC same SRC/DST
7	11	BACKDOOR Q access
8	1	(snort_decoder) WARNING: TCP Data Offset is less than 5!
9	5	BAD-TRAFFIC ip reserved bit set
10	1	SCAN nmap TCP
11	1	SCAN SOCKS Proxy attempt
12	1	SCAN Proxy Port 8080 attempt
13	1	SCAN Squid Proxy attempt
14	1	BARE BYTE UNICODE ENCODING
15	1	APACHE WHITESPACE (TAB)
16	1	BARE BYTE UNICODE ENCODING
17	3	NON-RFC HTTP DELIMITER (non-homogenous)
18	2	APACHE WHITESPACE (TAB) (non-homogenous)
19	2	APACHE WHITESPACE (TAB) (non-homogenous)
20	4	SCAN SOCKS Proxy attempt
21	4	SCAN Proxy Port 8080 attempt
22	4	SCAN Squid Proxy attempt
23	14	BAD-TRAFFIC tcp port 0 traffic

We found that each of these clusters is correct. Each cluster is one part (sometimes the only part) of an end user's interaction with the monitored network. This is precisely what we expect from combining this encoding with the autoassociator at this stage of our design. But, what is not obvious is that some of the clusters listed here are related, and should be signified as such to the IDS analyst. Looking closely at the data, we determined that these 23 clusters could optimally be collapsed into 10 clusters, if we were to combine correlated clusters. The following analysis helps motivate the foundations of our next design, which seeks to relate clusters of events with each other. If we are able to automatically slim down the number of clusters the IDS analyst must look at in this example to 10, we will be helping the analyst spend less time on false positives.

To see how the 23 clusters can be trimmed to 10, we note that one user is responsible for all the events in clusters 1-5, which all occur in a short period of time, even though there were significant numerical differences between our representations of the events. Different TCP flags are set on each of the packets, and different TCP sequence numbers and acknowledgement numbers are transmitted, among other numerical differences. Despite this, all 33 alerts occur in a one-minute window with source and destination IP addresses and TCP ports held static. So, it is likely that someone was simply transferring a file which contained many strings of `0x90` bytes.

As well, clusters 14-19 should be grouped together even though the source and destination ports and the destination address are not constant, since it appears that the threat was a non-standard web client connecting to an Apache web server.

Lastly, clusters 20-22 should be clustered together as one and, separately, clusters 10-12 should be clustered together as one. These two sequences of events are identical in order (showing they were probably produced by the same tool), but they happen 7 hours apart and do not come from the same IP address.

All of the other clusters seem to be independent of each other. The table shows how the autoassociator will perform sensitive clustering. The analysis given above offers some examples of what is expected of an algorithm that will find correlations between groups of events. We present the foundations of this algorithm in Section C.3.5 and Section C.3.6.

C.3.5 Event Cluster Correlation

Our final goal in creating clusters of related *Snort* events is to find correlation between these clusters and either individual *Snort* events of higher significance or other clusters of *Snort* events. When we accomplish this goal with some degree of success, if the link established is meaningful, we will be helping the IDS analyst in a very relevant way.

As another example of how this would help the IDS analyst, we present a specific scenario. It is common for a potential intruder to perform many reconnaissance probes before actually attempting a more damaging attack. We plan to develop a system that can help find correlation between different events like this attack and the recon probes.

For a hypothetical example of this common hacker technique, suppose a new buffer overflow exploit for the commonly used secure remote access tool *ssh* becomes public on the *Bugtraq* mailing list. In the days after this mailing list post, there is a sharp rise in recon probes originating from outside the firewall heading to TCP port 22 (the port *sshd* listens on by default) of different machines in the monitored network. By themselves, these probes are not necessarily damaging, so of greater interest to the IDS analyst is when one of these probes finds an open destination port on a machine in the lab and then a full connection to port 22 is established where the data transferred contains a string of bytes indicating a buffer overflow attempt (say: `0x90`, `0x90`, `...`, `0x90`, a sequence of `NOP` instructions on the common *Intel x86* architecture). Without a *Snort* event correlation engine, the IDS analyst will see a number of recon probes to port 22 and one buffer overflow alert (but these alerts are sometimes false alarms because strings of `0x90` are common in images) to the port hidden among all the other alerts generated at the time. The analyst might not connect the set of alerts to each other, so clearly if a tool were able to flag event correlations of this nature, this tool would be useful. At very least, a tool which found these correlations would save time for the analyst.

The system that is able to find correlations between clusters of events, as we did in the analysis of the last section, will solve the problem posed above as well.

Just as in the example from the last section, all the events over a set time window (one hour, one day, ...) will be pushed through the autoassociator to form a number of clusters. The recon probes from the hypothetical example would appear in the same cluster, and the buffer overflow alert would occur in a different cluster, possibly as the only item in that cluster. The system would be able to understand that the clusters are related, and would bring this to the attention of the IDS analyst.

The presence of a time window is important to avoid the problem of finding unimportant or spurious correlations. As well, by restricting the correlations which can be found to those available within a set time window, we will be dramatically reducing the size of the search space of possible correlations, as seen in last section's example where there are only 23 clusters to consider for a 10-hour window. By setting a time window, we are assuming that the attempted attack is most likely to occur near a set of recon probes. The danger in this scheme is that if this assumption is violated, then we might not allow the most correct correlation to be part of our search space. Of course, when it is obvious that important correlations are being missed because of the time window, it should be possible to make the time window larger to help solve the problem. And if we find this assumption of short temporal distance between related events is violated in general, we will revisit this design.

C.3.6 Group Correlations Implementation

The most central challenge in this problem is of producing meaningful correlations between groups of *Snort* events. In the Network Intrusion Detection: An Analyst's Handbook by Stephen Northcutt, the author presents some heuristics for solving this problem. Some of the most valuable methods of event correlation he mentions are: recognizing a common destination port (or range of destination ports), recognizing a common source IP address (or set of common source IP addresses), recognizing a common destination IP address, and recognizing related attack attempts through the constant difference in time between related attempts. The last method of correlation is the least obvious; basically, many attacks are performed by automatic tools, and because of this, there is often clear, systematic relationships between attempts made by the same run of the same tool (for example, the attack tool will try a candidate victim every 15 seconds, and this can be detected).

In the buffer exploit example we described in Section C.3.5, the clearest method of correlating the recon probes with the attack is through the common destination TCP port, port 22. But, in the example in Section C.3.4, the best way to correlate most of the clusters was by source IP address. In other cases still, it will be the time attribute which is most important. Another important consideration for us is distributed attacks. If a set of different IP addresses are probing and attempting intrusions on a server, we should be able to detect that (within reason – there will be some virtually untraceable attacks in any misuse detection system).

We will use neural networks to find correlations for this part of the problem as well, but where each cluster of *Snort* events is reconstituted as an individual data item. We do this to simplify the problem of finding the correlations, and to simplify the output of the system intuitively. But there is no very straightforward way to encode relationships like some of those listed above in a typical data item. What we wish to do is create a set of attributes for each data item which will be valuable in finding relationships when the data items are processed by a machine learning algorithm.

The first set of attributes we'll want to create is for set overlap of IP addresses. We mentioned that in any sort of distributed attack, the source IP address will not be constant, and we must deal with this. It in fact will be a set of source IP addresses that we're concerned with as the true source of the attack, and if we want to correlate this set with another set of IP addresses, we need to create an attribute where the amount of overlap between the sets can be used as a basis for comparison. To do this, we will create a list of all unique source IP addresses found in the time window we're working in. If this set of source IP addresses is small enough, we can create one attribute for each source IP address. If the list of unique source IP addresses is too long – so long that it will create data sparseness problems for our machine learning algorithm – we might consider creating a fixed set of attributes for all the source IP addresses, where each attribute represents a bucket of source IP addresses. To assign a value to these attributes for all data items, we will set the data item's value for this attribute to be 1 if any of the bucket's source IP addresses is present in the data item, and 0 otherwise. This construction would make it so that data items that have a similar set of source IP addresses would have a similar pattern of ones and zeroes for these bucket attributes. And, as long as there are enough bucket attributes, it is unlikely that two data items with very different sets of source IP addresses would have similar values for the bucket attributes, if a good hashing algorithm is used.

Clearly, this idea could be applied to both destination IP addresses and (source and destination) TCP ports in an intuitive manner.

Another idea for constructing attributes is creating two attributes to represent the time range over which all the *Snort* events in the data item took place. For each data item, we would find the time when the earliest event in the set happened and when the latest happened, then linearly map these values, relative to the start of the given time window, to the interval $[0, 1]$. These two mapped values would be the two new constructed attributes, representing the timeline of the data item.

We could also try this idea analogously for source and destination TCP ports, since there is a relationship between TCP ports immediately beside each other. (That is, if two TCP ports are numerically close to one another, they might both be part of the same service.)

These sets of attributes will cover what are generally understood to be the best factors in correlating seemingly disparate IDS alerts. Once we construct these sets of attributes, our task will be to find pairs or sets of data items, where each data item represents a cluster of similar events. If the performance of the system lacks after implementing these attributes, we will design different attributes or rethink our approach.

A final issue we'll note is that it will be difficult to test the strength of these higher-level correlations because of the *incidents.org* data set. Because the *incidents.org* data set is an unlabelled data set, it is very difficult to find stealthy intrusions which would evade our system. We plan to spend a good amount of time finding these hidden patterns by hand in the data – so that we can test our algorithms – but if we are not able to find a sufficient number of difficult patterns, we may be forced to look to other data sets (such as the *DARPA* data set) for difficult labelled attack attempts. The problem in adding elements from a different data set is that we might introduce uninteresting or structural trends into the new data set unwittingly.

C.4 April Report

Neural Network Application to Network Event Correlation

Progress Update

C.4.1 Overview of Report

In this report, we find that we are able to obtain better performance than reported in the last report by modifying the system previously preposed. We build on the last system, keeping most aspects of the system, but changing some important elements.

The latest system we propose offers a less dynamic clustering, thereby ameliorating the job of the IDS analyst. We have changed the system to produce less dynamic clusterings by training the neural network on a large number of alert examples.

When the neural network is trained on a large number of examples, the clusterings produced are more solid because their reconstruction error numbers do not change significantly when the set of testing examples changes, unlike the last system proposed.

Another change is that we no longer use explicit attributes from *Snort* (such as the *Snort* alert ID). This helps us show that our system is sufficiently general and that it could be used with non-*Snort* alerts or with input from other sources (such as routers or firewalls). This change did not affect the performance of the system noticeably, and we conclude that the *Snort*-generated attributes were not the primary features for clustering.

As well, the problem which our system solves has changed slightly as well. In the last report, we proposed a system which would try to find correlations between different alerts such that the expressed relationship represented some sort of higher semantic bond. For instance, we would have liked to cluster together all alerts generated by the same attacker within a set time period. We proposed to do this through a two-level system which would first group together alerts with very similar TCP, UDP, IP, or other protocol attributes, then discover further relationships between these groups using just the TCP port and IP address attributes. At both steps of this system, we used the autoassociator to produce reconstruction error values, and then used a very simple, ad hoc algorithm to produce the actual clusters. What we'd hoped to do with this system was first to group together similar types of alerts, and then to discover higher semantic bonds between these groups of alerts. We achieved this goal with some degree of success, but our goal has changed slightly to account for the original intent of this research project.

The goal of our research has changed in that we will no longer be looking for a higher semantic relationship between clusters of alerts. Instead, we will focus on simply grouping together similar types of alerts, so that the IDS analyst can decide which of these clusters of alerts are most important to watch as new alerts are reported in each group. This new system is effectively a way for an IDS analyst to prioritize all incoming alerts accurately so that he can focus on the important alerts rather than spending time analyzing false alarms. To achieve this new goal, we use a one-level clustering system, and pay more attention to tuning our system for optimal performance.

In Section C.4.2 of this report, we discuss the specifics of the new system we've introduced here. In Section C.4.3, we discuss the output from one run of our new algorithm. In the last section we summarize the conclusions of the analysis in Section C.4.3.

C.4.2 The Algorithm

The first step of the algorithm is transforming the *Snort* alerts into data items. This process is the same as in the last report, except that the attributes `snortGenId`, `snortSigId`, `snortSigRev`, `snortAlertClass`, `snortPriority`, `ipSrc`, and `ipDest` are no longer considered by the autoassociator. We not longer use the `snort*` attributes for generality, as previously mentioned, and we not longer use `ipSrc` and `ipDest` because we have found that these attributes hinder rather than help correct clustering.

The next step of the algorithm, where the autoassociator is used on constructed data items, changes only slightly. The description of the autoassociator is the same as in the last report, except we note that the parameter values for the number of hidden units and the number of training epochs seem to be optimal at 8 hidden units and 500 epochs. Although we note this, we have not proved the optimality of these values, and we share them as a way to reconstruct our research more than as a scientific result.

As we mentioned in the overview of this report, we also train the autoassociator on a number of data instances before doing the actual test clustering. For this report, we reserved the first 1000 *Snort* alerts of the *incidents.org* data set as testing data, though we only use 200 of these reserved alerts in our analysis, for brevity. We used the first 1000 alerts rather than the last 1000 because we found that the alerts in this initial set are more diverse and interesting than most other random samplings of 1000 sequential alerts. We wanted diversity in our testing data set to illustrate with more rigour the performance of our scheme. We trained the autoassociator on the next 10,000 *Snort* alerts of the *incidents.org* data set, after the initial 1000 alerts. This differs from the previous report, because in the previous report we used the autoassociator for clustering without first training it. We added this long training step to stabilize the values of the reconstruction errors produced by the autoassociator.

C.4.3 Analysis of Output

In this section we analyze the output of our algorithm. We analyze our output in depth to help illustrate how our algorithm functions, and how it may be improved. We ran the algorithm on the *incidents.org* data set (as specified in the previous section) and our tools produced the textual output seen in the appendix (see Section C.5) to this report. (Essentially, what we are doing in this section is commenting on the output of our clustering program when it is told to cluster 200 interesting alerts.) We include the appendix in full so that we may refer to it often, without quoting large sections of it.

The first two clusters in the output contain four alerts total, all `http_inspect` alerts, all produced by TCP sessions with an *Apache* web server. The alerts in the first cluster are both produced by the same session. (We can tell this because the IP addresses and TCP port, sequence and acknowledgement numbers match up very closely. As well, the packets were received within one second of each other.) The alerts in the second cluster are both from a different connection which took place 4 seconds earlier than the connection from the first cluster. (We know that the alerts in second cluster are from the same session for the same reasons we saw for the first cluster, and we know that these two clusters of alerts are from different TCP sessions because the TCP port numbers are different.

These first two clusters spell out the trend we will see for the rest of this analysis. We see that the way we are using the autoassociator for this problem creates a high level of numerical sensitivity so that it is possible to distinguish between groups of alerts with subtle differences. The alerts from the first cluster are very similar to the alerts of the second cluster, but the numerical differences in the TCP acknowledgement and sequence number attributes are enough to make the autoassociator notice that the sets of alerts are distinct.

Moving on to the third cluster, we have 23 `SHELLCODE x86 NOOP` alerts, all originating in the same TCP session, over a span of about 12 seconds. If an administrator were to see all of these alerts clustered together like this, it would be very simple for him to see that these alerts were probably caused by a large data file transfer which contained a number of strings of `0x90` bytes. This is an example of how this tool might be useful if it were developed further.

The fourth cluster contains a continuation of the third cluster, in the form of 12 more `SHELLCODE` alerts. These alerts are clearly from the same data connection as the alerts from the third cluster, but these alerts occur 15 seconds later than the previous group. All 12 of these alerts are for packets from within a 0.25 second time window, so they are all very similar numerically. The reason the third cluster and fourth cluster are not grouped together as one cluster (as they logically should be) is the numerical differences in the TCP acknowledgement number. This number increases as the data connection matures, and so a consequence of the numerical sensitivity of our scheme is that if a particular attribute varies too widely, the logical cluster will be broken into two or more clusters.

The fifth cluster is quite simple. It contains 3 alerts for one `Apache` session, which are all related and nicely self-contained.

The sixth cluster is very similar to the `SHELLCODE x86 NOOP` alerts from the third cluster. In fact, between these two clusters, every attribute shares a very similar range, save one. The IP TOS (*type of service*) attribute has the value `0xA0` for every alert in clusters 3 and 4, but every alert in cluster 6 has the value `0x00` for the TOS attribute. This is another example of the numerical sensitivity.

Next cluster, cluster 7, contains only one alert, which looks similar to the alerts in cluster 4. The difference between cluster 4 and cluster 7 is the alert in cluster 7 took place 6 hours before the alert in cluster 4. As such, the TCP sequence numbers are very different between the clusters.

Cluster 8 and 11 each contain one isolated alert which doesn't seem to relate to any others in the test set. Cluster 9 and 10 contain one alert each, and these two alerts probably should be clustered together. The alerts are identical in every attribute considered by the autoassociator, except for the IP ID attribute. The alerts differ by about 2000 in this attribute, and clearly this was a large enough number for the autoassociator to consider the alerts separate. Looking at the other cohesive groups of alerts, the IP ID attribute never seems to differ by a number as large as this, so clearly the autoassociator understood something subtle that an IDS analyst might not have picked up.

Cluster 12 contains 4 alerts which ostensibly seem to correlate neatly. But with cluster 13, things become more interesting again. In this cluster, there are 10 `SCAN nmap TCP`, all focused on port 80, targeting 3 different IP addresses. These alerts originate from 6 different IP addresses, and it seems clear that someone was trying to cover their tracks while probing port 80. These *nmap* scans are all numerically similar, especially in the TCP sequence number attribute. As well, in this cluster there are 8 scans of TCP ports for *Squid* and *SOCKS* – both programs relating to proxying HTTP data. There are 4 attempts on each port, and they are all done by the same IP address in a short period of time. These scans are related to the 4 scans in cluster 12 (they are done by the same IP address in a short period of time), which tried a different HTTP proxy port: 8080. Clearly these 12 alerts are related.

Although there are groups of related alerts within this cluster, the *nmap* scans are not related to the HTTP proxy scans. The numerical difference between the two sets of alerts is very great. The reason these alerts have been clustered together lies in a fundamental problem with our scheme which we'd like to address. When we compute the reconstruction error on the autoassociator outputs, we are essentially collapsing a more than 40 dimensional number into a 1 dimensional number, so that we can perform a numerical comparison between all the data items easily. As we see with this example, this scheme is problematic. Any formula which provides a mapping that removes dimensionality will lose information, but the problem is particularly acute in our case because more than 40 dimensions are being mapped to a single dimension for the purpose of clustering.

We believe that this problem is the most significant factor in holding back the performance of our correlation engine. We do not present a solution to this problem now, so we continue with the analysis of the algorithm output. We should also note that this problem is not as significant as it may sound at first. A main goal of our research has been to produce close-to-deterministic mappings for alerts of similar types. If unrelated alerts are being mapped to similar reconstruction errors, it does compromise the practical usability of the scheme somewhat, but it doesn't compromise our goal of providing deterministic mappings.

The next cluster, cluster 15, is another example of an *Apache* encoding alert. The two alerts in this cluster are obviously related and warrant no further inspection. The next cluster contains 5 *nmap* scans of port 80. They are all to and from the same IP addresses, so this cluster is accurate as well.

Cluster 17 is more interesting. It contains 34 alerts, all of type `BACKDOOR Q access` (and there are no other clusters containing this type of alert). The source IP address on all these packets is the broadcast address (255.255.255.255) and the source TCP port is 31337 (which phonetically spells "elite", a common adjective among malicious hackers). All of the destination IP addresses are different, but they are also all within the same class B IP network. From the point of the autoassociator, these alerts are all identical, so they are all grouped together.

Clusters 18, 19, and 20 are all related *Apache* web server warnings which do not warrant further discussion.

Cluster 21 is similar to cluster 13 in that cluster 21 is composed of both *nmap* scans and unrelated HTTP proxy scans. The next cluster contains five alerts which look like they should be in cluster 21, 4 *nmap* alerts and a *Squid* probe. The other 9 alerts in the cluster are very similar (but very different from the *nmap* alerts, numerically).

These 9 alerts are all of type `BAD-TRAFFIC ip reserved bit set`, sent to different IP addresses, so there was probably an attacker scanning for machines on the network by using malformed packets.

Cluster 23 contains 7 `SHELLCODE x86 NOOP` which look like they should be in cluster 3 (they are from the same session as those in cluster 3), but their IP ID and TCP sequence numbers differ enough to set them apart. This cluster also has an amalgam of other miscellaneous packets in it, such as the detection of a *GNUTella* client – a peer-to-peer file sharing program – and 2 more (related) *nmap* scans. One trend we see in our system is that as reconstruction error rates rise, there is often more noise in the clusterings we produce. We explain this by noting that the autoassociator computes higher reconstruction errors for packets it has seen less, so the autoassociator is likely less tuned to the nuances of these infrequent packets. Cluster 24 is comprised of a *GNUTella* packet clearly related to the one in cluster 23. Cluster 25 holds only an *nmap* scan alert seemingly unrelated to the other *nmap* scans.

Cluster 26 has two related malformed TCP packets, and cluster 27 has one malformed TCP packets which is related to the two in cluster 26. Cluster 27 also has 9 illegal TCP packets (with TCP port 0), which are all related. It also has a `SHELLCODE` alert which should be in cluster 4.

Clusters 28, 29, 30, and 31 all have 1 alert each, an illegal TCP packet alert. They are all related to the illegal TCP packets in cluster 26 because they all have only the TCP reset bit set, and they all have their TCP sequence and acknowledgement number set to be equal. These alerts are not showing up in the same cluster because this sequence/acknowledgement number is random for each packet, and so there is too much variation in these attributes.

Clusters 32, 33, and 34 have only 1 alert each, an HTTP proxy scan, and they are related, so they should have been grouped together.

The last interesting cluster is cluster 35, which contains only `BAD-TRAFFIC same SRC/DST` alerts (which indicate a packet had the same source and destination IP addresses). This cluster contains 10 alerts (all of the alerts of that type) so it is accurate. The last 3 clusters contain only one (uninteresting) alert each.

C.4.4 Summary of Output Analysis

The most significant conclusion to note is that the mapping of the autoassociator outputs done by the reconstruction error formula is not *one-to-one*, because it cannot be. This problem crops up in a few cases, as we saw with this analysis of the output.

Another important conclusion to note is that for the most part, the mappings we generate are deterministic and that alerts which are clustered close to one another have a greater chance of being related. For instance, although clusters 1 and 2 are separated by a cluster barrier, it is clear from their cluster ordinals that these two clusters are more likely to be related than clusters 2 and 20. This property arises from the way the reconstruction error formula works, and this is the property we explicitly exploit in our simple clustering algorithm.

Another property of this clustering scheme to note is that the clusterings are sensitive to subtle numerical differences between the encodings of the data instances. From this we can conclude that it may be worthwhile to experiment with tuning the data instance encodings (further than what we've already done) and the way the autoassociator works with them.

The final property we mentioned is the property which says that, in general, as the reconstruction error grows, so does the level of noise in the clusterings we produce. This property is interesting, amd could be incorporating into a production system. We could rank alerts with higher reconstruction error rates as being more important for the IDS analyst to look at.

C.5 Appendix to April Report

Neural Network Application to Network Event Correlation

Appendix to April Report

This appendix is the output of one run of our system on 200 *Snort* alerts generated from the *incidents.org* [1] data set.

```
Cluster 1:  (3.2575)
--------------------
[**][119:4:1] (http_inspect) BARE BYTE UNICODE ENCODING [**]
11/15-05:01:26.376507 170.129.50.120:63188 -> 207.68.164.250:80
TCP TTL:125 TOS:0x0 ID:5826 IpLen:20 DgmLen:932 DF
**AP*** Seq:0xC9F1EEBB Ack:0xF019024A Win:0xFA61 TcpLen:20


[**][119:12:1] (http_inspect) APACHE WHITESPACE (TAB) [**]
11/15-05:01:25.396507 170.129.50.120:63188 -> 207.68.164.250:80
TCP TTL:125 TOS:0x0 ID:5777 IpLen:20 DgmLen:932 DF
**AP*** Seq:0xC9F10EBB Ack:0xF019024A Win:0xFA61 TcpLen:20


Cluster 2:  (3.2883)
--------------------
[**][119:4:1] (http_inspect) BARE BYTE UNICODE ENCODING [**]
11/15-05:01:21.766507 170.129.50.120:63175 -> 207.68.164.250:80
TCP TTL:125 TOS:0x0 ID:5651 IpLen:20 DgmLen:932 DF
**AP*** Seq:0xC9E10861 Ack:0x3015901B Win:0xFA61 TcpLen:20


[**][119:12:1] (http_inspect) APACHE WHITESPACE (TAB) [**]
11/15-05:01:20.766507 170.129.50.120:63175 -> 207.68.164.250:80
TCP TTL:125 TOS:0x0 ID:5602 IpLen:20 DgmLen:932 DF
**AP*** Seq:0xC9E02861 Ack:0x3015901B Win:0xFA61 TcpLen:20

Cluster 3:  (3.3431)
--------------------
[**][1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:55:36.566507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:46490 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA074E240 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
```

```
[Xref=> http://www.whitehats.com/info/IDS181]

[**][1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority: 1]
11/14-21:55:36.576507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:46491 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA074E7A4 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
[Xref=> http://www.whitehats.com/info/IDS181]

[**][1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority: 1]
11/14-21:55:36.756507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:46504 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA0752DB8 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
[Xref=> http://www.whitehats.com/info/IDS181]

[**][1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority: 1]
11/14-21:55:37.146507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:46529 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA075B47C Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
[Xref=> http://www.whitehats.com/info/IDS181]

[**][1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority: 1]
11/14-21:55:37.216507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:46532 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA075C4A8 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
[Xref=> http://www.whitehats.com/info/IDS181]

[**][1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority: 1]
11/14-21:55:37.526507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:46553 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA07635DC Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
[Xref=> http://www.whitehats.com/info/IDS181]

[**][1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority: 1]
11/14-21:55:37.626507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:46560 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA0765B98 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
```

```
[Xref=> http://www.whitehats.com/info/IDS181]

[**][1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:55:37.856507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:46575 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA076AC74 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
[Xref=> http://www.whitehats.com/info/IDS181]

[**][1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:55:37.876507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:46577 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA076B73C Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
[Xref=> http://www.whitehats.com/info/IDS181]

[**][1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:55:38.016507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:46584 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA076DCF8 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
[Xref=> http://www.whitehats.com/info/IDS181]

[**][1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:55:38.936507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:46646 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA0782B30 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
[Xref=> http://www.whitehats.com/info/IDS181]

[**][1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:55:39.306507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:46667 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA0789C64 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
[Xref=> http://www.whitehats.com/info/IDS181]

[**][1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:55:39.436507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:46673 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA078BCBC Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
```

```
[Xref=> http://www.whitehats.com/info/IDS181]

[**][1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:55:40.176507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:46719 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA079B4B4 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
[Xref=> http://www.whitehats.com/info/IDS181]

[**][1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:55:47.876507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:47208 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA08400B8 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
[Xref=> http://www.whitehats.com/info/IDS181]

[**][1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:55:48.206507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:47231 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA0847CB4 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
[Xref=> http://www.whitehats.com/info/IDS181]

[**][1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:55:48.226507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:47233 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA084877C Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
[Xref=> http://www.whitehats.com/info/IDS181]

[**][1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:55:48.456507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:47245 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA084C82C Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
[Xref=> http://www.whitehats.com/info/IDS181]

[**][1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:55:48.466507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:47247 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA084D2F4 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
```

```
[Xref=> http://www.whitehats.com/info/IDS181]

[**][1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:55:48.516507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:47251 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA084E884 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
[Xref=> http://www.whitehats.com/info/IDS181]

[**][1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:55:48.596507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:47256 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA0850378 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
[Xref=> http://www.whitehats.com/info/IDS181]

Cluster 4:  (3.3496)
--------------------
[**][1:1390:4] SHELLCODE x86 inc ebx NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:56:03.796507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:48293 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA09AD98C Ack:0x90CF9E29 Win:0x16D0 TcpLen:20

[**][1:1390:4] SHELLCODE x86 inc ebx NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:56:03.806507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:48294 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA09ADEF0 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20

[**][1:1390:4] SHELLCODE x86 inc ebx NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:56:03.816507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:48295 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA09AE454 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20

[**][1:1390:4] SHELLCODE x86 inc ebx NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:56:03.836507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:48296 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA09AE9B8 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
```

```
[**][1:1390:4] SHELLCODE x86 inc ebx NOOP [**]
[Classification:Executable code was detected] [Priority: 1]
11/14-21:56:03.846507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:48297 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA09AEF1C Ack:0x90CF9E29 Win:0x16D0 TcpLen:20

[**][1:1390:4] SHELLCODE x86 inc ebx NOOP [**]
[Classification:Executable code was detected] [Priority: 1]
11/14-21:56:03.856507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:48299 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA09AF9E4 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20

[**][1:1390:4] SHELLCODE x86 inc ebx NOOP [**]
[Classification:Executable code was detected] [Priority: 1]
11/14-21:56:03.886507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:48300 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA09AFF48 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20

[**][1:1390:4] SHELLCODE x86 inc ebx NOOP [**]
[Classification:Executable code was detected] [Priority: 1]
11/14-21:56:03.896507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:48301 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA09B04AC Ack:0x90CF9E29 Win:0x16D0 TcpLen:20

[**][1:1390:4] SHELLCODE x86 inc ebx NOOP [**]
[Classification:Executable code was detected] [Priority: 1]
11/14-21:56:03.916507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:48303 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA09B0F74 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20

[**][1:1390:4] SHELLCODE x86 inc ebx NOOP [**]
[Classification:Executable code was detected] [Priority: 1]
11/14-21:56:03.936507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:48304 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA09B14D8 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20

[**][1:1390:4] SHELLCODE x86 inc ebx NOOP [**]
[Classification:Executable code was detected] [Priority: 1]
11/14-21:56:03.946507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:48305 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA09B1A3C Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
```

```
[**][1:1390:4] SHELLCODE x86 inc ebx NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:56:03.966507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:48306 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA09B1FA0 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20

Cluster 5:  (3.3603)
--------------------
[**][119:13:1] (http_inspect) NON-RFC HTTP DELIMITER [**]
11/15-04:45:24.366507 170.129.50.120:61044 -> 216.136.173.111:80
TCP TTL:125 TOS:0x0 ID:10198 IpLen:20 DgmLen:932 DF
**AP*** Seq:0xA014995 Ack:0x6193010 Win:0xFAF0 TcpLen:20

[**][119:12:1] (http_inspect) APACHE WHITESPACE (TAB) [**]
11/15-04:45:21.306507 170.129.50.120:61044 -> 216.136.173.111:80
TCP TTL:125 TOS:0x0 ID:10054 IpLen:20 DgmLen:932 DF
**AP*** Seq:0x9FEC995 Ack:0x6193010 Win:0xFAF0 TcpLen:20

[**][119:4:1] (http_inspect) BARE BYTE UNICODE ENCODING [**]
11/15-04:45:11.546507 170.129.50.120:61044 -> 216.136.173.111:80
TCP TTL:125 TOS:0x0 ID:9584 IpLen:20 DgmLen:932 DF
**AP*** Seq:0x9F68995 Ack:0x6193010 Win:0xFAF0 TcpLen:20

Cluster 6:  (3.3765)
--------------------
[**][1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:55:36.676507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0x0 ID:46498 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA0750D60 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
[Xref=> http://www.whitehats.com/info/IDS181]

[**][1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:55:37.026507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0x0 ID:46521 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA075895C Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
[Xref=> http://www.whitehats.com/info/IDS181]

[**][1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:55:39.356507 129.118.2.10:57425 -> 170.129.50.120:63414
```

```
TCP TTL:51 TOS:0x0 ID:46669 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA078A72C Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
[Xref=> http://www.whitehats.com/info/IDS181]

[**][1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:55:48.316507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0x0 ID:47239 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA084A7D4 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
[Xref=> http://www.whitehats.com/info/IDS181]

[**][1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:55:48.506507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0x0 ID:47250 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA084E320 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
[Xref=> http://www.whitehats.com/info/IDS181]

[**][1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:55:48.576507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0x0 ID:47255 IpLen:20 DgmLen:1420 DF
**A**** Seq:0xA084FE14 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
[Xref=> http://www.whitehats.com/info/IDS181]

Cluster 7:  (3.3918)
--------------------
[**][1:1390:4] SHELLCODE x86 inc ebx NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-16:10:30.806507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:56986 IpLen:20 DgmLen:1420 DF
**A**** Seq:0x8217BFFC Ack:0x90CF9E29 Win:0x16D0 TcpLen:20

Cluster 8:  (3.4744)
--------------------
[**][119:4:1] (http_inspect) BARE BYTE UNICODE ENCODING [**]
11/14-13:03:42.666507 170.129.50.120:63598 -> 64.4.22.250:80
TCP TTL:124 TOS:0x0 ID:56064 IpLen:20 DgmLen:932 DF
**AP*** Seq:0x94851318 Ack:0x9AB18054 Win:0x43E1 TcpLen:20

Cluster 9:  (3.4798)
--------------------
```

```
[**][1:618:5] SCAN Squid Proxy attempt [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-16:06:24.316507 206.48.61.139:4006 -> 170.129.23.239:3128
TCP TTL:114 TOS:0x0 ID:26758 IpLen:20 DgmLen:48 DF
*****S* Seq:  0x13D84F7 Ack:  0x0 Win:  0x2000 TcpLen:  28
TCP Options (4) => MSS: 536 NOP NOP SackOK

Cluster 10:  (3.4854)
--------------------
[**][1:618:5] SCAN Squid Proxy attempt [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-16:06:21.366507 206.48.61.139:4006 -> 170.129.23.239:3128
TCP TTL:114 TOS:0x0 ID:24710 IpLen:20 DgmLen:48 DF
*****S* Seq:  0x13D84F7 Ack:  0x0 Win:  0x2000 TcpLen:  28
TCP Options (4) => MSS: 536 NOP NOP SackOK

Cluster 11:  (3.4906)
--------------------
[**][116:46:1] (snort_decoder) WARNING: TCP Data Offset is less
 than 5!  [**]
11/14-22:53:21.926507 68.41.28.138:0 -> 170.129.225.41:0
TCP TTL:107 TOS:0x0 ID:35590 IpLen:20 DgmLen:48 DF
******* Seq:  0x50989C Ack:  0x2F470000 Win:  0x7002 TcpLen:  0

Cluster 12:  (3.5082)
--------------------
[**][1:620:6] SCAN Proxy Port 8080 attempt [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/15-07:00:08.07650 216.201.160.235:44389->170.129.161.213:8080
TCP TTL:46 TOS:0x0 ID:52415 IpLen:20 DgmLen:60 DF
*****S* Seq:  0x4E70998D Ack:  0x0 Win:  0x16D0 TcpLen:  40
TCP Options (5) => MSS: 1460 SackOK TS: 17306018 0 NOP WS: 0

[**][1:620:6] SCAN Proxy Port 8080 attempt [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/15-06:59:56.08650 216.201.160.235:44389->170.129.161.213:8080
TCP TTL:46 TOS:0x0 ID:52197 IpLen:20 DgmLen:60 DF
*****S* Seq:  0x4E70998D Ack:  0x0 Win:  0x16D0 TcpLen:  40
TCP Options (5) => MSS: 1460 SackOK TS: 17304818 0 NOP WS: 0

[**][1:620:6] SCAN Proxy Port 8080 attempt [**]
[Classification:Attempted Information Leak] [Priority:  2]
```

```
11/15-06:59:50.11650 216.201.160.235:44389->170.129.161.213:8080
TCP TTL:46 TOS:0x0 ID:52135 IpLen:20 DgmLen:60 DF
*****S* Seq:  0x4E70998D Ack:  0x0 Win:  0x16D0 TcpLen:  40
TCP Options (5) => MSS: 1460 SackOK TS: 17304218 0 NOP WS: 0

[**][1:620:6] SCAN Proxy Port 8080 attempt [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/15-06:59:47.09650 216.201.160.235:44389->170.129.161.213:8080
TCP TTL:46 TOS:0x0 ID:52064 IpLen:20 DgmLen:60 DF
*****S* Seq:  0x4E70998D Ack:  0x0 Win:  0x16D0 TcpLen:  40
TCP Options (5) => MSS: 1460 SackOK TS: 17303918 0 NOP WS: 0

Cluster 13:  (3.5147)
--------------------
[**][1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-12:14:46.366507 210.66.117.5:80 -> 170.129.81.112:80
TCP TTL:47 TOS:0x0 ID:51968 IpLen:20 DgmLen:40
**A**** Seq:  0x400 Ack:  0x0 Win:  0x578 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS28]

[**][1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-12:14:26.276507 61.222.192.98:80 -> 170.129.81.112:80
TCP TTL:49 TOS:0x0 ID:49820 IpLen:20 DgmLen:40
**A**** Seq:  0x271 Ack:  0x0 Win:  0x578 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS28]

[**][1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-12:14:41.366507 210.66.117.5:80 -> 170.129.81.112:80
TCP TTL:47 TOS:0x0 ID:51450 IpLen:20 DgmLen:40
**A**** Seq:  0x39E Ack:  0x0 Win:  0x578 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS28]

[**][1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-12:14:21.266507 61.222.192.98:80 -> 170.129.81.112:80
TCP TTL:49 TOS:0x0 ID:49258 IpLen:20 DgmLen:40
**A**** Seq:  0x20A Ack:  0x0 Win:  0x578 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS28]
```

```
[**][1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-12:14:16.266507 61.222.14.98:80 -> 170.129.81.112:80
TCP TTL:49 TOS:0x0 ID:48734 IpLen:20 DgmLen:40
**A**** Seq:  0x1A8 Ack:  0x0 Win:  0x578 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS28]

[**][1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-12:14:11.266507 61.222.14.98:80 -> 170.129.81.112:80
TCP TTL:49 TOS:0x0 ID:48192 IpLen:20 DgmLen:40
**A**** Seq:  0x144 Ack:  0x0 Win:  0x578 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS28]

[**][1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-14:50:18.166507 61.218.161.202:80 -> 170.129.151.28:80
TCP TTL:48 TOS:0x0 ID:64698 IpLen:20 DgmLen:40
**A**** Seq:  0x26 Ack:  0x0 Win:  0x578 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS28]

[**][1:618:5] SCAN Squid Proxy attempt [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/15-07:00:08.04650 216.201.160.235:44394->170.129.161.213:3128
TCP TTL:46 TOS:0x0 ID:52420 IpLen:20 DgmLen:60 DF
*****S* Seq:  0x4EE97664 Ack:  0x0 Win:  0x16D0 TcpLen:  40
TCP Options (5) => MSS: 1460 SackOK TS: 17306018 0 NOP WS: 0

[**][1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-14:50:23.136507 61.218.161.202:80 -> 170.129.151.28:80
TCP TTL:48 TOS:0x0 ID:64986 IpLen:20 DgmLen:40
**A**** Seq:  0x8C Ack:  0x0 Win:  0x578 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS28]

[**][1:618:5] SCAN Squid Proxy attempt [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/15-06:59:56.07650 216.201.160.235:44394->170.129.161.213:3128
TCP TTL:46 TOS:0x0 ID:52202 IpLen:20 DgmLen:60 DF
*****S* Seq:  0x4EE97664 Ack:  0x0 Win:  0x16D0 TcpLen:  40
TCP Options (5) => MSS: 1460 SackOK TS: 17304818 0 NOP WS: 0
```

```
[**][1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority: 2]
11/14-14:50:28.126507 61.218.161.210:80 -> 170.129.151.28:80
TCP TTL:48 TOS:0x0 ID:65271 IpLen:20 DgmLen:40
**A**** Seq: 0x100 Ack: 0x0 Win: 0x578 TcpLen: 20
[Xref=> http://www.whitehats.com/info/IDS28]

[**][1:618:5] SCAN Squid Proxy attempt [**]
[Classification:Attempted Information Leak] [Priority: 2]
11/15-06:59:50.12650 216.201.160.235:44394->170.129.161.213:3128
TCP TTL:46 TOS:0x0 ID:52140 IpLen:20 DgmLen:60 DF
*****S* Seq: 0x4EE97664 Ack: 0x0 Win: 0x16D0 TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 17304218 0 NOP WS: 0

[**][1:618:5] SCAN Squid Proxy attempt [**]
[Classification:Attempted Information Leak] [Priority: 2]
11/15-06:59:47.09650 216.201.160.235:44394->170.129.161.213:3128
TCP TTL:46 TOS:0x0 ID:52069 IpLen:20 DgmLen:60 DF
*****S* Seq: 0x4EE97664 Ack: 0x0 Win: 0x16D0 TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 17303918 0 NOP WS: 0

[**][1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority: 2]
11/14-23:18:23.756507 61.222.251.82:80 -> 170.129.50.3:80
TCP TTL:48 TOS:0x0 ID:44368 IpLen:20 DgmLen:40
**A**** Seq: 0x6D Ack: 0x0 Win: 0x578 TcpLen: 20
[Xref=> http://www.whitehats.com/info/IDS28]

[**][1:615:5] SCAN SOCKS Proxy attempt [**]
[Classification:Attempted Information Leak] [Priority: 2]
11/15-07:00:08.07650 216.201.160.235:44398->170.129.161.213:1080
TCP TTL:46 TOS:0x0 ID:52424 IpLen:20 DgmLen:60 DF
*****S* Seq: 0x4E410469 Ack: 0x0 Win: 0x16D0 TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 17306018 0 NOP WS: 0
[Xref=> http://help.undernet.org/proxyscan/]

[**][1:615:5] SCAN SOCKS Proxy attempt [**]
[Classification:Attempted Information Leak] [Priority: 2]
11/15-06:59:56.08650 216.201.160.235:44398->170.129.161.213:1080
TCP TTL:46 TOS:0x0 ID:52206 IpLen:20 DgmLen:60 DF
*****S* Seq: 0x4E410469 Ack: 0x0 Win: 0x16D0 TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 17304818 0 NOP WS: 0
```

```
[Xref=> http://help.undernet.org/proxyscan/]

[**][1:615:5] SCAN SOCKS Proxy attempt [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/15-06:59:50.11650 216.201.160.235:44398->170.129.161.213:1080
TCP TTL:46 TOS:0x0 ID:52144 IpLen:20 DgmLen:60 DF
*****S* Seq:  0x4E410469 Ack:  0x0 Win:  0x16D0 TcpLen:  40
TCP Options (5) => MSS: 1460 SackOK TS: 17304218 0 NOP WS: 0
[Xref=> http://help.undernet.org/proxyscan/]

[**][1:615:5] SCAN SOCKS Proxy attempt [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/15-06:59:47.09650 216.201.160.235:44398->170.129.161.213:1080
TCP TTL:46 TOS:0x0 ID:52073 IpLen:20 DgmLen:60 DF
*****S* Seq:  0x4E410469 Ack:  0x0 Win:  0x16D0 TcpLen:  40
TCP Options (5) => MSS: 1460 SackOK TS: 17303918 0 NOP WS: 0
[Xref=> http://help.undernet.org/proxyscan/]

Cluster 14:  (3.5404)
--------------------
[**][1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-10:10:23.856507 163.23.238.9:80 -> 170.129.19.170:80
TCP TTL:44 TOS:0x0 ID:31290 IpLen:20 DgmLen:40
**A**** Seq:  0x300 Ack:  0x0 Win:  0x578 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS28]

[**][1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-10:10:18.856507 61.218.161.210:80 -> 170.129.19.170:80
TCP TTL:48 TOS:0x0 ID:30943 IpLen:20 DgmLen:40
**A**** Seq:  0x278 Ack:  0x0 Win:  0x578 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS28]

[**][1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-10:10:13.826507 61.218.161.210:80 -> 170.129.19.170:80
TCP TTL:48 TOS:0x0 ID:30662 IpLen:20 DgmLen:40
**A**** Seq:  0x20A Ack:  0x0 Win:  0x578 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS28]

[**][1:628:3] SCAN nmap TCP [**]
```

```
[Classification:Attempted Information Leak] [Priority: 2]
11/14-10:10:08.786507 61.218.161.202:80 -> 170.129.19.170:80
TCP TTL:48 TOS:0x0 ID:30366 IpLen:20 DgmLen:40
**A**** Seq: 0x198 Ack: 0x0 Win: 0x578 TcpLen: 20
[Xref=> http://www.whitehats.com/info/IDS28]

[**][1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority: 2]
11/14-10:10:03.816507 61.218.161.202:80 -> 170.129.19.170:80
TCP TTL:48 TOS:0x0 ID:30084 IpLen:20 DgmLen:40
**A**** Seq: 0x134 Ack: 0x0 Win: 0x578 TcpLen: 20
[Xref=> http://www.whitehats.com/info/IDS28]

[**][1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority: 2]
11/14-12:52:09.906507 163.23.238.9:80 -> 170.129.14.62:80
TCP TTL:44 TOS:0x0 ID:29544 IpLen:20 DgmLen:40
**A**** Seq: 0x2FD Ack: 0x0 Win: 0x578 TcpLen: 20
[Xref=> http://www.whitehats.com/info/IDS28]

[**][1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority: 2]
11/14-12:52:04.916507 61.218.161.210:80 -> 170.129.14.62:80
TCP TTL:48 TOS:0x0 ID:29197 IpLen:20 DgmLen:40
**A**** Seq: 0x279 Ack: 0x0 Win: 0x578 TcpLen: 20
[Xref=> http://www.whitehats.com/info/IDS28]

[**][1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority: 2]
11/14-12:51:59.916507 61.218.161.210:80 -> 170.129.14.62:80
TCP TTL:48 TOS:0x0 ID:28911 IpLen:20 DgmLen:40
**A**** Seq: 0x209 Ack: 0x0 Win: 0x578 TcpLen: 20
[Xref=> http://www.whitehats.com/info/IDS28]

[**][1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority: 2]
11/14-12:51:54.916507 61.218.161.202:80 -> 170.129.14.62:80
TCP TTL:48 TOS:0x0 ID:28601 IpLen:20 DgmLen:40
**A**** Seq: 0x18D Ack: 0x0 Win: 0x578 TcpLen: 20
[Xref=> http://www.whitehats.com/info/IDS28]

[**][1:628:3] SCAN nmap TCP [**]
```

```
[Classification:Attempted Information Leak] [Priority:  2]
11/14-12:51:49.906507 61.218.161.202:80 -> 170.129.14.62:80
TCP TTL:48 TOS:0x0 ID:28299 IpLen:20 DgmLen:40
**A**** Seq:  0x11B Ack:  0x0 Win:  0x578 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS28]

Cluster 15:  (3.5510)
--------------------
[**][119:4:1] (http_inspect) BARE BYTE UNICODE ENCODING [**]
11/14-09:54:38.316507 170.129.50.120:63362 -> 159.153.199.24:80
TCP TTL:125 TOS:0x0 ID:38908 IpLen:20 DgmLen:436 DF
**AP*** Seq:0x99AD8FC7 Ack:0x732FBFCD Win:0x4230 TcpLen:20

[**][119:4:1] (http_inspect) BARE BYTE UNICODE ENCODING [**]
11/14-09:54:47.286507 170.129.50.120:63387 -> 159.153.199.24:80
TCP TTL:125 TOS:0x0 ID:38984 IpLen:20 DgmLen:436 DF
**AP*** Seq:0x99C8B003 Ack:0x733D8EE2 Win:0x4230 TcpLen:20

Cluster 16:  (3.5562)
--------------------
[**][1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-14:29:34.136507 202.29.28.1:80 -> 170.129.238.112:80
TCP TTL:45 TOS:0x0 ID:23947 IpLen:20 DgmLen:40
**A**** Seq:  0x5D Ack:  0x0 Win:  0x578 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS28]

[**][1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-14:29:24.186507 202.29.28.1:80 -> 170.129.238.112:80
TCP TTL:45 TOS:0x0 ID:23770 IpLen:20 DgmLen:40
**A**** Seq:  0x0 Ack:  0x0 Win:  0x578 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS28]

[**][1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-14:29:14.306507 202.29.28.1:80 -> 170.129.238.112:80
TCP TTL:45 TOS:0x0 ID:23584 IpLen:20 DgmLen:40
**A**** Seq:  0x3A6 Ack:  0x0 Win:  0x578 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS28]

[**][1:628:3] SCAN nmap TCP [**]
```

```
[Classification:Attempted Information Leak] [Priority:  2]
11/14-14:29:04.266507 202.29.28.1:80 -> 170.129.238.112:80
TCP TTL:45 TOS:0x0 ID:23404 IpLen:20 DgmLen:40
**A**** Seq:  0x348 Ack:  0x0 Win:  0x578 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS28]

[**][1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-14:28:54.256507 202.29.28.1:80 -> 170.129.238.112:80
TCP TTL:45 TOS:0x0 ID:23204 IpLen:20 DgmLen:40
**A**** Seq:  0x2E6 Ack:  0x0 Win:  0x578 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS28]

Cluster 17:  (3.5706)
--------------------
[**][1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority:  3]
11/14-09:29:14.82650 255.255.255.255:31337->170.129.172.186:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq:  0x0 Ack:  0x0 Win:  0x0 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS203]

[**][1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority:  3]
11/14-09:32:53.01650 255.255.255.255:31337->170.129.132.79:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq:  0x0 Ack:  0x0 Win:  0x0 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS203]

[**][1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority:  3]
11/14-09:49:26.15650 255.255.255.255:31337->170.129.129.188:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq:  0x0 Ack:  0x0 Win:  0x0 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS203]

[**][1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority:  3]
11/14-10:10:22.59650 255.255.255.255:31337->170.129.195.178:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq:  0x0 Ack:  0x0 Win:  0x0 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS203]
```

```
[**][1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority: 3]
11/14-10:44:04.83650 255.255.255.255:31337->170.129.30.34:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq: 0x0 Ack: 0x0 Win: 0x0 TcpLen: 20
[Xref=> http://www.whitehats.com/info/IDS203]

[**][1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority: 3]
11/14-11:44:56.15650 255.255.255.255:31337->170.129.137.174:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq: 0x0 Ack: 0x0 Win: 0x0 TcpLen: 20
[Xref=> http://www.whitehats.com/info/IDS203]

[**][1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority: 3]
11/14-12:56:26.74650 255.255.255.255:31337->170.129.89.87:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq: 0x0 Ack: 0x0 Win: 0x0 TcpLen: 20
[Xref=> http://www.whitehats.com/info/IDS203]

[**][1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority: 3]
11/14-14:01:20.98650 255.255.255.255:31337->170.129.23.133:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq: 0x0 Ack: 0x0 Win: 0x0 TcpLen: 20
[Xref=> http://www.whitehats.com/info/IDS203]

[**][1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority: 3]
11/14-14:04:08.96650 255.255.255.255:31337->170.129.190.188:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq: 0x0 Ack: 0x0 Win: 0x0 TcpLen: 20
[Xref=> http://www.whitehats.com/info/IDS203]

[**][1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority: 3]
11/14-16:14:32.96650 255.255.255.255:31337->170.129.192.22:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq: 0x0 Ack: 0x0 Win: 0x0 TcpLen: 20
[Xref=> http://www.whitehats.com/info/IDS203]
```

```
[**][1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority: 3]
11/14-16:16:56.98650 255.255.255.255:31337->170.129.134.5:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq: 0x0 Ack: 0x0 Win: 0x0 TcpLen: 20
[Xref=> http://www.whitehats.com/info/IDS203]

[**][1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority: 3]
11/14-16:17:12.03650 255.255.255.255:31337->170.129.156.132:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq: 0x0 Ack: 0x0 Win: 0x0 TcpLen: 20
[Xref=> http://www.whitehats.com/info/IDS203]

[**][1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority: 3]
11/14-16:30:57.08650 255.255.255.255:31337->170.129.146.62:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq: 0x0 Ack: 0x0 Win: 0x0 TcpLen: 20
[Xref=> http://www.whitehats.com/info/IDS203]

[**][1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority: 3]
11/14-16:47:18.15650 255.255.255.255:31337->170.129.176.42:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq: 0x0 Ack: 0x0 Win: 0x0 TcpLen: 20
[Xref=> http://www.whitehats.com/info/IDS203]

[**][1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority: 3]
11/14-17:06:30.60650 255.255.255.255:31337->170.129.80.5:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq: 0x0 Ack: 0x0 Win: 0x0 TcpLen: 20
[Xref=> http://www.whitehats.com/info/IDS203]

[**][1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority: 3]
11/14-17:11:03.61650 255.255.255.255:31337->170.129.1.102:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq: 0x0 Ack: 0x0 Win: 0x0 TcpLen: 20
[Xref=> http://www.whitehats.com/info/IDS203]
```

```
[**][1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority: 3]
11/14-17:23:24.64650 255.255.255.255:31337->170.129.41.171:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq: 0x0 Ack: 0x0 Win: 0x0 TcpLen: 20
[Xref=> http://www.whitehats.com/info/IDS203]

[**][1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority: 3]
11/14-19:34:10.59650 255.255.255.255:31337->170.129.94.129:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq: 0x0 Ack: 0x0 Win: 0x0 TcpLen: 20
[Xref=> http://www.whitehats.com/info/IDS203]

[**][1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority: 3]
11/14-19:38:10.75650 255.255.255.255:31337->170.129.181.145:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq: 0x0 Ack: 0x0 Win: 0x0 TcpLen: 20
[Xref=> http://www.whitehats.com/info/IDS203]

[**][1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority: 3]
11/14-20:56:56.23650 255.255.255.255:31337->170.129.72.205:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq: 0x0 Ack: 0x0 Win: 0x0 TcpLen: 20
[Xref=> http://www.whitehats.com/info/IDS203]

[**][1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority: 3]
11/14-21:10:50.47650 255.255.255.255:31337->170.129.156.91:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq: 0x0 Ack: 0x0 Win: 0x0 TcpLen: 20
[Xref=> http://www.whitehats.com/info/IDS203]

[**][1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority: 3]
11/14-21:29:40.69650 255.255.255.255:31337->170.129.161.211:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq: 0x0 Ack: 0x0 Win: 0x0 TcpLen: 20
[Xref=> http://www.whitehats.com/info/IDS203]
```

```
[**][1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority:  3]
11/14-22:50:02.01650 255.255.255.255:31337->170.129.103.3:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq:  0x0 Ack:  0x0 Win:  0x0 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS203]

[**][1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority:  3]
11/14-13:35:08.89650 255.255.255.255:31337->170.129.200.84:515
TCP TTL:14 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq:  0x0 Ack:  0x0 Win:  0x0 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS203]

[**][1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority:  3]
11/14-23:47:53.41650 255.255.255.255:31337->170.129.19.28:515
TCP TTL:14 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq:  0x0 Ack:  0x0 Win:  0x0 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS203]

[**][1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority:  3]
11/14-23:55:50.45650 255.255.255.255:31337->170.129.161.133:515
TCP TTL:14 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq:  0x0 Ack:  0x0 Win:  0x0 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS203]

[**][1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority:  3]
11/15-00:38:36.89650 255.255.255.255:31337->170.129.57.163:515
TCP TTL:14 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq:  0x0 Ack:  0x0 Win:  0x0 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS203]

[**][1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority:  3]
11/15-03:00:36.04650 255.255.255.255:31337->170.129.193.103:515
TCP TTL:14 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq:  0x0 Ack:  0x0 Win:  0x0 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS203]
```

```
[**][1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority:  3]
11/15-04:16:09.18650 255.255.255.255:31337->170.129.53.148:515
TCP TTL:14 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq:  0x0 Ack:  0x0 Win:  0x0 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS203]

[**][1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority:  3]
11/15-04:36:09.28650 255.255.255.255:31337->170.129.178.16:515
TCP TTL:14 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq:  0x0 Ack:  0x0 Win:  0x0 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS203]

[**][1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority:  3]
11/15-04:52:48.24650 255.255.255.255:31337->170.129.155.128:515
TCP TTL:14 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq:  0x0 Ack:  0x0 Win:  0x0 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS203]

[**][1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority:  3]
11/15-07:17:35.84650 255.255.255.255:31337->170.129.153.135:515
TCP TTL:14 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq:  0x0 Ack:  0x0 Win:  0x0 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS203]

[**][1:184:4] BACKDOOR Q access [**]
[Classification:Misc activity] [Priority:  3]
11/15-07:22:02.82650 255.255.255.255:31337->170.129.65.138:515
TCP TTL:14 TOS:0x0 ID:0 IpLen:20 DgmLen:43
**A*R** Seq:  0x0 Ack:  0x0 Win:  0x0 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS203]

Cluster 18:  (3.5735)
--------------------
[**][119:4:1] (http_inspect) BARE BYTE UNICODE ENCODING [**]
11/15-02:46:26.726507 170.129.50.120:64749 -> 216.130.211.11:80
TCP TTL:124 TOS:0x0 ID:26873 IpLen:20 DgmLen:1332 DF
**AP*** Seq:0x1600D507 Ack:0xF2FBCCD5 Win:0x2058 TcpLen:20
```

```
Cluster 19:  (3.5769)
--------------------
[**][119:4:1] (http_inspect) BARE BYTE UNICODE ENCODING [**]
11/15-02:48:02.606507 170.129.50.120:64868 -> 216.130.211.11:80
TCP TTL:124 TOS:0x0 ID:29178 IpLen:20 DgmLen:1332 DF
**AP*** Seq:0x1772D6D5 Ack:0x4E05CFF6 Win:0x2058 TcpLen:20

Cluster 20:  (3.5900)
--------------------
[**][119:12:1] (http_inspect) APACHE WHITESPACE (TAB) [**]
11/15-02:46:28.446507 170.129.50.120:64749 -> 216.130.211.11:80
TCP TTL:124 TOS:0x0 ID:40697 IpLen:20 DgmLen:1332 DF
**AP*** Seq:0x1601F507 Ack:0xF2FBCCD5 Win:0x2058 TcpLen:20

Cluster 21:  (3.5962)
--------------------
[**][1:620:6] SCAN Proxy Port 8080 attempt [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-16:00:56.996507 66.159.18.66:43517 -> 170.129.50.120:8080
TCP TTL:53 TOS:0x0 ID:59575 IpLen:20 DgmLen:60 DF
*****S* Seq:  0xBED8745 Ack:  0x0 Win:  0x16D0 TcpLen:  40
TCP Options (5) => MSS: 1460 SackOK TS: 48656370 0 NOP WS: 0

[**][1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-12:02:46.046507 167.79.91.3:80 -> 170.129.50.122:53
TCP TTL:47 TOS:0x0 ID:11664 IpLen:20 DgmLen:40
**A**** Seq:  0x2A7 Ack:  0x0 Win:  0x578 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS28]

[**][1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-12:02:45.976507 167.79.91.3:80 -> 170.129.50.122:53
TCP TTL:49 TOS:0x0 ID:11661 IpLen:20 DgmLen:40
**A**** Seq:  0x2A5 Ack:  0x0 Win:  0x578 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS28]

[**][1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-20:34:04.926507 192.192.171.251:80 -> 170.129.69.49:80
```

```
TCP TTL:44 TOS:0x0 ID:11428 IpLen:20 DgmLen:40
**A**** Seq:  0x353 Ack:  0x0 Win:  0x578 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS28]

[**][1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-20:34:00.056507 192.192.171.251:80 -> 170.129.69.49:80
TCP TTL:44 TOS:0x0 ID:10868 IpLen:20 DgmLen:40
**A**** Seq:  0x2EA Ack:  0x0 Win:  0x578 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS28]

[**][1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-20:33:54.676507 61.221.88.198:80 -> 170.129.69.49:80
TCP TTL:50 TOS:0x0 ID:10358 IpLen:20 DgmLen:40
**A**** Seq:  0x286 Ack:  0x0 Win:  0x578 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS28]

Cluster 22:  (3.6055)
--------------------
[**][1:618:5] SCAN Squid Proxy attempt [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-16:00:56.996507 66.159.18.66:43518 -> 170.129.50.120:3128
TCP TTL:53 TOS:0x0 ID:50174 IpLen:20 DgmLen:60 DF
******S* Seq:  0xBF3AC0C Ack:  0x0 Win:  0x16D0 TcpLen:  40
TCP Options (5) => MSS: 1460 SackOK TS: 48656370 0 NOP WS: 0

[**][1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-20:33:44.536507 61.218.15.126:80 -> 170.129.69.49:80
TCP TTL:50 TOS:0x0 ID:9312 IpLen:20 DgmLen:40
**A**** Seq:  0x1BF Ack:  0x0 Win:  0x578 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS28]

[**][1:523:4] BAD-TRAFFIC ip reserved bit set [**]
[Classification:Misc activity] [Priority:  3]
11/14-23:28:20.656507 200.200.200.1 -> 170.129.53.47
TCP TTL:241 TOS:0x0 ID:0 IpLen:20 DgmLen:40 RB
Frag Offset:  0x0864 Frag Size:  0x0014

[**][1:523:4] BAD-TRAFFIC ip reserved bit set [**]
[Classification:Misc activity] [Priority:  3]
```

```
11/15-01:09:54.346507 200.200.200.1 -> 170.129.217.111
TCP TTL:241 TOS:0x0 ID:0 IpLen:20 DgmLen:40 RB
Frag Offset:  0x0864 Frag Size:  0x0014

[**][1:523:4] BAD-TRAFFIC ip reserved bit set [**]
[Classification:Misc activity] [Priority:  3]
11/15-01:21:09.846507 200.200.200.1 -> 170.129.127.227
TCP TTL:241 TOS:0x0 ID:0 IpLen:20 DgmLen:40 RB
Frag Offset:  0x0864 Frag Size:  0x0014

[**][1:523:4] BAD-TRAFFIC ip reserved bit set [**]
[Classification:Misc activity] [Priority:  3]
11/15-05:24:51.026507 200.200.200.1 -> 170.129.60.231
TCP TTL:241 TOS:0x0 ID:0 IpLen:20 DgmLen:40 RB
Frag Offset:  0x0864 Frag Size:  0x0014

[**][1:523:4] BAD-TRAFFIC ip reserved bit set [**]
[Classification:Misc activity] [Priority:  3]
11/14-11:21:09.916507 200.200.200.1 -> 170.129.211.200
TCP TTL:242 TOS:0x0 ID:0 IpLen:20 DgmLen:40 RB
Frag Offset:  0x0864 Frag Size:  0x0014

[**][1:523:4] BAD-TRAFFIC ip reserved bit set [**]
[Classification:Misc activity] [Priority:  3]
11/14-14:37:18.296507 200.200.200.1 -> 170.129.2.16
TCP TTL:242 TOS:0x0 ID:0 IpLen:20 DgmLen:40 RB
Frag Offset:  0x0864 Frag Size:  0x0014

[**][1:523:4] BAD-TRAFFIC ip reserved bit set [**]
[Classification:Misc activity] [Priority:  3]
11/14-15:54:39.456507 200.200.200.1 -> 170.129.79.180
TCP TTL:242 TOS:0x0 ID:0 IpLen:20 DgmLen:40 RB
Frag Offset:  0x0864 Frag Size:  0x0014

[**][1:523:4] BAD-TRAFFIC ip reserved bit set [**]
[Classification:Misc activity] [Priority:  3]
11/14-17:59:31.346507 200.200.200.1 -> 170.129.239.44
TCP TTL:242 TOS:0x0 ID:0 IpLen:20 DgmLen:40 RB
Frag Offset:  0x0864 Frag Size:  0x0014

[**][1:523:4] BAD-TRAFFIC ip reserved bit set [**]
[Classification:Misc activity] [Priority:  3]
```

```
11/14-23:04:22.106507 200.200.200.1 -> 170.129.43.122
TCP TTL:242 TOS:0x0 ID:0 IpLen:20 DgmLen:40 RB
Frag Offset: 0x0864 Frag Size: 0x0014

[**][1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority: 2]
11/14-20:33:39.636507 61.218.15.126:80 -> 170.129.69.49:80
TCP TTL:50 TOS:0x0 ID:8768 IpLen:20 DgmLen:40
**A**** Seq: 0x158 Ack: 0x0 Win: 0x578 TcpLen: 20
[Xref=> http://www.whitehats.com/info/IDS28]

[**][1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority: 2]
11/14-20:33:34.876507 61.218.15.118:80 -> 170.129.69.49:80
TCP TTL:50 TOS:0x0 ID:8252 IpLen:20 DgmLen:40
**A**** Seq: 0xF4 Ack: 0x0 Win: 0x578 TcpLen: 20
[Xref=> http://www.whitehats.com/info/IDS28]

[**][1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority: 2]
11/14-20:33:29.606507 61.218.15.118:80 -> 170.129.69.49:80
TCP TTL:50 TOS:0x0 ID:7722 IpLen:20 DgmLen:40
**A**** Seq: 0x92 Ack: 0x0 Win: 0x578 TcpLen: 20
[Xref=> http://www.whitehats.com/info/IDS28]

Cluster 23: (3.6383)
--------------------
[**][1:615:5] SCAN SOCKS Proxy attempt [**]
[Classification:Attempted Information Leak] [Priority: 2]
11/14-23:32:20.686507 66.159.18.49:55989 -> 170.129.50.120:1080
TCP TTL:52 TOS:0x0 ID:59203 IpLen:20 DgmLen:60 DF
*****S* Seq: 0xB4F124E7 Ack: 0x0 Win: 0x16D0 TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 51364772 0 NOP WS: 0
[Xref=> http://help.undernet.org/proxyscan/]

[**][1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority: 1]
11/14-21:55:37.296507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:46538 IpLen:20 DgmLen:1420 DF
**AP*** Seq:0xA075E500 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
[Xref=> http://www.whitehats.com/info/IDS181]
```

```
[**][1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:55:37.866507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:46576 IpLen:20 DgmLen:1420 DF
**AP*** Seq:0xA076B1D8 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
[Xref=> http://www.whitehats.com/info/IDS181]

[**][1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:55:41.096507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:46776 IpLen:20 DgmLen:1420 DF
**AP*** Seq:0xA07AE7F8 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
[Xref=> http://www.whitehats.com/info/IDS181]

[**][1:620:6] SCAN Proxy Port 8080 attempt [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-23:32:20.806507 66.159.18.49:55990 -> 170.129.50.120:8080
TCP TTL:52 TOS:0x0 ID:52037 IpLen:20 DgmLen:60 DF
******S* Seq:  0xB4F108EA Ack:  0x0 Win:  0x16D0 TcpLen:  40
TCP Options (5) => MSS: 1460 SackOK TS: 51364783 0 NOP WS: 0

[**][1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:55:47.826507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:47206 IpLen:20 DgmLen:1420 DF
**AP*** Seq:0xA083F5F0 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
[Xref=> http://www.whitehats.com/info/IDS181]

[**][1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:55:48.306507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:47238 IpLen:20 DgmLen:1420 DF
**AP*** Seq:0xA084A270 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
[Xref=> http://www.whitehats.com/info/IDS181]

[**][1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:55:48.476507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:47248 IpLen:20 DgmLen:1420 DF
**AP*** Seq:0xA084D858 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
[Xref=> http://www.whitehats.com/info/IDS181]
```

```
[**][1:648:6] SHELLCODE x86 NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:55:48.536507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:47252 IpLen:20 DgmLen:1420 DF
**AP*** Seq:0xA084EDE8 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20
[Xref=> http://www.whitehats.com/info/IDS181]

[**][1:556:5] P2P Outbound GNUTella client request [**]
[Classification:Potential Corporate Privacy Violation]
[Priority:  1]
11/14-15:43:37.096507 170.129.50.120:61121 -> 24.65.114.32:6003
TCP TTL:123 TOS:0x0 ID:22720 IpLen:20 DgmLen:158 DF
**AP*** Seq:0x5A0CA92C Ack:0x53A65413 Win:0x4038 TcpLen:20

[**][1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-14:50:38.206507 163.23.238.9:80 -> 170.129.151.28:80
TCP TTL:44 TOS:0x0 ID:336 IpLen:20 DgmLen:40
**A**** Seq:  0x1EC Ack:  0x0 Win:  0x578 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS28]

[**][1:1390:4] SHELLCODE x86 inc ebx NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:56:03.906507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:48302 IpLen:20 DgmLen:1420 DF
**AP*** Seq:0xA09B0A10 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20

[**][1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-14:50:33.266507 61.218.161.210:80 -> 170.129.151.28:80
TCP TTL:48 TOS:0x0 ID:32 IpLen:20 DgmLen:40
**A**** Seq:  0x17A Ack:  0x0 Win:  0x578 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS28]

Cluster 24:  (3.6706)
--------------------
[**][1:556:5] P2P Outbound GNUTella client request [**]
[Classification:Potential Corporate Privacy Violation]
[Priority:  1]
11/14-15:43:37.316507 170.129.50.120:61122 -> 24.65.114.32:6003
TCP TTL:123 TOS:0x0 ID:22766 IpLen:20 DgmLen:62 DF
**AP*** Seq:0x5A129557 Ack:0x53A7A3BA Win:0x4038 TcpLen:20
```

```
Cluster 25:  (3.6798)
--------------------
[**][1:628:3] SCAN nmap TCP [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-10:52:22.116507 63.211.17.228:80 -> 170.129.50.120:63874
TCP TTL:54 TOS:0x0 ID:563 IpLen:20 DgmLen:40
**A**** Seq:  0x3DC Ack:  0x0 Win:  0x578 TcpLen:  20
[Xref=> http://www.whitehats.com/info/IDS28]

Cluster 26:  (3.6883)
--------------------
[**][116:46:1] (snort_decoder) WARNING: TCP Data Offset is less
 than 5!  [**]
[Classification:Potential Corporate Privacy Violation]
 [Priority:  1]
11/10-01:12:17.866507 172.20.10.199:0 -> 207.166.119.62:0
TCP TTL:235 TOS:0x0 ID:0 IpLen:20 DgmLen:40
****R** Seq:  0xBDD2D468 Ack:  0xBDD2D468 Win:  0x0 TcpLen:  16

[**][116:46:1] (snort_decoder) WARNING: TCP Data Offset is less
 than 5!  [**]
[Classification:Potential Corporate Privacy Violation]
 [Priority:  1]
11/11-09:08:23.926507 172.20.10.199:0 -> 207.166.207.98:0
TCP TTL:234 TOS:0x0 ID:0 IpLen:20 DgmLen:40
****R** Seq:  0xC4AD19A6 Ack:  0xC4AD19A6 Win:  0x0 TcpLen:  16

Cluster 27:  (3.7100)
--------------------
[**][116:46:1] (snort_decoder) WARNING: TCP Data Offset is
 less than 5!  [**]
[Classification:Potential Corporate Privacy Violation]
 [Priority:  1]
11/10-07:20:11.976507 172.20.10.199:0 -> 207.166.168.10:0
TCP TTL:235 TOS:0x0 ID:0 IpLen:20 DgmLen:40
****R** Seq:  0xBF23A8C4 Ack:  0xBF23A8C4 Win:  0x0 TcpLen:  0

[**][1:524:7] BAD-TRAFFIC tcp port 0 traffic [**]
[Classification:Misc activity] [Priority:  3]
11/15-07:34:50.446507 211.47.255.24:41104 -> 170.129.195.40:0
```

```
TCP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
*****S* Seq:  0xD30F0032 Ack:  0x0 Win:  0x16D0 TcpLen:  32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

[**][1:524:7] BAD-TRAFFIC tcp port 0 traffic [**]
[Classification:Misc activity] [Priority:  3]
11/15-07:34:53.296507 211.47.255.24:41104 -> 170.129.195.40:0
TCP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
*****S* Seq:  0xD30F0032 Ack:  0x0 Win:  0x16D0 TcpLen:  32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

[**][1:524:7] BAD-TRAFFIC tcp port 0 traffic [**]
[Classification:Misc activity] [Priority:  3]
11/15-07:34:59.466507 211.47.255.24:41104 -> 170.129.195.40:0
TCP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
*****S* Seq:  0xD30F0032 Ack:  0x0 Win:  0x16D0 TcpLen:  32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

[**][1:524:7] BAD-TRAFFIC tcp port 0 traffic [**]
[Classification:Misc activity] [Priority:  3]
11/15-07:35:11.276507 211.47.255.24:41104 -> 170.129.195.40:0
TCP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
*****S* Seq:  0xD30F0032 Ack:  0x0 Win:  0x16D0 TcpLen:  32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

[**][1:524:7] BAD-TRAFFIC tcp port 0 traffic [**]
[Classification:Misc activity] [Priority:  3]
11/15-07:35:22.326507 211.47.255.24:41358 -> 170.129.195.40:0
TCP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
*****S* Seq:  0xD4C4A857 Ack:  0x0 Win:  0x16D0 TcpLen:  32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

[**][1:524:7] BAD-TRAFFIC tcp port 0 traffic [**]
[Classification:Misc activity] [Priority:  3]
11/15-07:35:25.406507 211.47.255.24:41358 -> 170.129.195.40:0
TCP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
*****S* Seq:  0xD4C4A857 Ack:  0x0 Win:  0x16D0 TcpLen:  32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

[**][1:524:7] BAD-TRAFFIC tcp port 0 traffic [**]
[Classification:Misc activity] [Priority:  3]
11/15-07:35:31.326507 211.47.255.24:41358 -> 170.129.195.40:0
```

```
TCP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
*****S* Seq:  0xD4C4A857 Ack:  0x0 Win:  0x16D0 TcpLen:  32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

[**][1:524:7] BAD-TRAFFIC tcp port 0 traffic [**]
[Classification:Misc activity] [Priority:  3]
11/15-07:35:54.326507 211.47.255.24:41611 -> 170.129.195.40:0
TCP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
*****S* Seq:  0xD60E9D41 Ack:  0x0 Win:  0x16D0 TcpLen:  32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

[**][1:524:7] BAD-TRAFFIC tcp port 0 traffic [**]
[Classification:Misc activity] [Priority:  3]
11/15-07:35:57.316507 211.47.255.24:41611 -> 170.129.195.40:0
TCP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
*****S* Seq:  0xD60E9D41 Ack:  0x0 Win:  0x16D0 TcpLen:  32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

[**][1:524:7] BAD-TRAFFIC tcp port 0 traffic [**]
[Classification:Misc activity] [Priority:  3]
11/15-07:36:03.286507 211.47.255.24:41611 -> 170.129.195.40:0
TCP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
*****S* Seq:  0xD60E9D41 Ack:  0x0 Win:  0x16D0 TcpLen:  32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

[**][1:524:7] BAD-TRAFFIC tcp port 0 traffic [**]
[Classification:Misc activity] [Priority:  3]
11/15-07:36:15.296507 211.47.255.24:41611 -> 170.129.195.40:0
TCP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
*****S* Seq:  0xD60E9D41 Ack:  0x0 Win:  0x16D0 TcpLen:  32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

[**][1:524:7] BAD-TRAFFIC tcp port 0 traffic [**]
[Classification:Misc activity] [Priority:  3]
11/15-07:36:26.406507 211.47.255.24:41866 -> 170.129.195.40:0
TCP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
*****S* Seq:  0xD8010CF5 Ack:  0x0 Win:  0x16D0 TcpLen:  32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

[**][1:524:7] BAD-TRAFFIC tcp port 0 traffic [**]
[Classification:Misc activity] [Priority:  3]
11/15-07:36:29.296507 211.47.255.24:41866 -> 170.129.195.40:0
```

```
TCP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
*****S* Seq:  0xD8010CF5 Ack:  0x0 Win:  0x16D0 TcpLen:  32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

[**][1:524:7] BAD-TRAFFIC tcp port 0 traffic [**]
[Classification:Misc activity] [Priority:  3]
11/15-07:36:35.286507 211.47.255.24:41866 -> 170.129.195.40:0
TCP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
*****S* Seq:  0xD8010CF5 Ack:  0x0 Win:  0x16D0 TcpLen:  32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

[**][1:1390:4] SHELLCODE x86 inc ebx NOOP [**]
[Classification:Executable code was detected] [Priority:  1]
11/14-21:56:03.856507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0x0 ID:48298 IpLen:20 DgmLen:1420 DF
**AP*** Seq:0xA09AF480 Ack:0x90CF9E29 Win:0x16D0 TcpLen:20

Cluster 28:  (3.7198)
--------------------
[**][116:46:1] (snort_decoder) WARNING: TCP Data Offset is less
 than 5!  [**]
[Classification:Potential Corporate Privacy Violation]
 [Priority:  1]
11/12-03:07:49.886507 210.243.145.141:0 -> 207.166.159.139:0
TCP TTL:237 TOS:0x0 ID:0 IpLen:20 DgmLen:40
****R** Seq:0x158662C0 Ack:0x158662C0 Win:0x0 TcpLen:16

Cluster 29:  (3.7297)
--------------------
[**][116:46:1] (snort_decoder) WARNING: TCP Data Offset is less
 than 5!  [**]
[Classification:Potential Corporate Privacy Violation]
 [Priority:  1]
11/10-05:38:56.936507 62.13.27.29:0 -> 207.166.25.86:0
TCP TTL:234 TOS:0x0 ID:0 IpLen:20 DgmLen:40
****R** Seq:  0xA02C678 Ack:  0xA02C678 Win:  0x0 TcpLen:  8

Cluster 30:  (3.7446)
--------------------
[**][116:46:1] (snort_decoder) WARNING: TCP Data Offset is less
 than 5!  [**]
[Classification:Potential Corporate Privacy Violation]
```

```
 [Priority:  1]
11/09-20:51:11.676507 62.13.27.29:0 -> 207.166.33.145:0
TCP TTL:234 TOS:0x0 ID:0 IpLen:20 DgmLen:40
****R** Seq:  0x81F9750 Ack:  0x81F9750 Win:  0x0 TcpLen:  0

Cluster 31:  (3.7529)
--------------------
[**][116:46:1] (snort_decoder) WARNING: TCP Data Offset is less
 than 5!  [**]
[Classification:Potential Corporate Privacy Violation]
 [Priority:  1]
11/12-20:25:11.826507 217.209.183.235:0 -> 207.166.252.249:0
TCP TTL:236 TOS:0x0 ID:0 IpLen:20 DgmLen:40
****R** Seq:  0x24A1C4C Ack:  0x24A1C4C Win:  0x0 TcpLen:  0

Cluster 32:  (3.7738)
--------------------
[**][1:615:5] SCAN SOCKS Proxy attempt [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-16:00:56.996507 66.159.18.66:43520 -> 170.129.50.120:1080
TCP TTL:53 TOS:0x0 ID:4253 IpLen:20 DgmLen:60 DF
*****S* Seq:  0xB9A7F6F Ack:  0x0 Win:  0x16D0 TcpLen:  40
TCP Options (5) => MSS: 1460 SackOK TS: 48656370 0 NOP WS: 0
[Xref=> http://help.undernet.org/proxyscan/]

Cluster 33:  (3.7825)
--------------------
[**][1:615:5] SCAN SOCKS Proxy attempt [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-16:00:56.996507 66.159.18.66:43521 -> 170.129.50.120:1080
TCP TTL:53 TOS:0x0 ID:2662 IpLen:20 DgmLen:60 DF
*****S* Seq:  0xBF46594 Ack:  0x0 Win:  0x16D0 TcpLen:  40
TCP Options (5) => MSS: 1460 SackOK TS: 48656370 0 NOP WS: 0
[Xref=> http://help.undernet.org/proxyscan/]

Cluster 34:  (3.7967)
--------------------
[**][1:618:5] SCAN Squid Proxy attempt [**]
[Classification:Attempted Information Leak] [Priority:  2]
11/14-23:32:20.916507 66.159.18.49:55991 -> 170.129.50.120:3128
TCP TTL:52 TOS:0x0 ID:16353 IpLen:20 DgmLen:60 DF
*****S* Seq:  0xB4E1FC5B Ack:  0x0 Win:  0x16D0 TcpLen:  40
```

```
TCP Options (5) => MSS: 1460 SackOK TS: 51364794 0 NOP WS: 0

Cluster 35:  (3.8518)
--------------------
[**][1:527:4] BAD-TRAFFIC same SRC/DST [**]
[Classification:Potentially Bad Traffic] [Priority:  2]
11/14-22:36:45.306507 170.129.215.99 -> 170.129.215.99
IGMP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:28
[Xref=> http://www.cert.org/advisories/CA-1997-28.html]

[**][1:527:4] BAD-TRAFFIC same SRC/DST [**]
[Classification:Potentially Bad Traffic] [Priority:  2]
11/14-22:36:45.306507 170.129.215.104 -> 170.129.215.104
IGMP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:28
[Xref=> http://www.cert.org/advisories/CA-1997-28.html]

[**][1:527:4] BAD-TRAFFIC same SRC/DST [**]
[Classification:Potentially Bad Traffic] [Priority:  2]
11/14-22:36:45.306507 170.129.215.115 -> 170.129.215.115
IGMP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:28
[Xref=> http://www.cert.org/advisories/CA-1997-28.html]

[**][1:527:4] BAD-TRAFFIC same SRC/DST [**]
[Classification:Potentially Bad Traffic] [Priority:  2]
11/14-22:36:45.306507 170.129.215.110 -> 170.129.215.110
IGMP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:28
[Xref=> http://www.cert.org/advisories/CA-1997-28.html]

[**][1:527:4] BAD-TRAFFIC same SRC/DST [**]
[Classification:Potentially Bad Traffic] [Priority:  2]
11/14-22:36:45.306507 170.129.215.120 -> 170.129.215.120
IGMP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:28
[Xref=> http://www.cert.org/advisories/CA-1997-28.html]

[**][1:527:4] BAD-TRAFFIC same SRC/DST [**]
[Classification:Potentially Bad Traffic] [Priority:  2]
11/14-22:36:45.306507 170.129.215.126 -> 170.129.215.126
IGMP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:28
[Xref=> http://www.cert.org/advisories/CA-1997-28.html]

[**][1:527:4] BAD-TRAFFIC same SRC/DST [**]
[Classification:Potentially Bad Traffic] [Priority:  2]
```

```
11/14-22:36:45.306507 170.129.215.131 -> 170.129.215.131
IGMP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:28
[Xref=> http://www.cert.org/advisories/CA-1997-28.html]

[**][1:527:4] BAD-TRAFFIC same SRC/DST [**]
[Classification:Potentially Bad Traffic] [Priority: 2]
11/14-22:36:45.306507 170.129.215.137 -> 170.129.215.137
IGMP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:28
[Xref=> http://www.cert.org/advisories/CA-1997-28.html]

[**][1:527:4] BAD-TRAFFIC same SRC/DST [**]
[Classification:Potentially Bad Traffic] [Priority: 2]
11/14-22:36:45.306507 170.129.215.142 -> 170.129.215.142
IGMP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:28
[Xref=> http://www.cert.org/advisories/CA-1997-28.html]

[**][1:527:4] BAD-TRAFFIC same SRC/DST [**]
[Classification:Potentially Bad Traffic] [Priority: 2]
11/14-22:36:45.306507 170.129.215.85 -> 170.129.215.85
IGMP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:28
[Xref=> http://www.cert.org/advisories/CA-1997-28.html]

[**][1:527:4] BAD-TRAFFIC same SRC/DST [**]
[Classification:Potentially Bad Traffic] [Priority: 2]
11/14-22:36:45.306507 170.129.215.93 -> 170.129.215.93
IGMP TTL:47 TOS:0x0 ID:0 IpLen:20 DgmLen:28
[Xref=> http://www.cert.org/advisories/CA-1997-28.html]

Cluster 36: (3.8629)
--------------------
[**][116:46:1] (snort_decoder) WARNING: TCP Data Offset is less
 than 5! [**]
[Classification:Potential Corporate Privacy Violation]
 [Priority: 1]
11/10-01:28:34.556507 62.13.27.29:0 -> 207.166.78.44:0
TCP TTL:234 TOS:0x0 ID:0 IpLen:20 DgmLen:40 DF
****R** Seq: 0x91D8C02 Ack: 0x91D8C02 Win: 0x0 TcpLen: 12

Cluster 37: (3.8955)
--------------------
[**][116:46:1] (snort_decoder) WARNING: TCP Data Offset is
 less than 5! [**]
```

```
[Classification:Potential Corporate Privacy Violation]
 [Priority:  1]
11/12-18:38:17.846507 203.80.239.162:0 -> 207.166.182.137:0
TCP TTL:107 TOS:0x0 ID:35119 IpLen:20 DgmLen:48 DF
1*UA**** Seq:0x7930005 Ack:0xD80A04D1 Win:0x64BA TcpLen:0
 UrgPtr:0x800

Cluster 38:  (4.0391)
--------------------
[**][116:97:1] (snort_decoder):  Short UDP packet, length
 field > payload length [**]
[Classification:Potential Corporate Privacy Violation]
 [Priority:  1]
11/11-13:29:54.796507 211.194.68.39:0 -> 207.166.72.218:0
UDP TTL:109 TOS:0x0 ID:2062 IpLen:20 DgmLen:78
Len:  129
```

C.6 DARPA Test Data

The following is a set from the DARPA test data for 200000 epochs and 0.003 MSE.

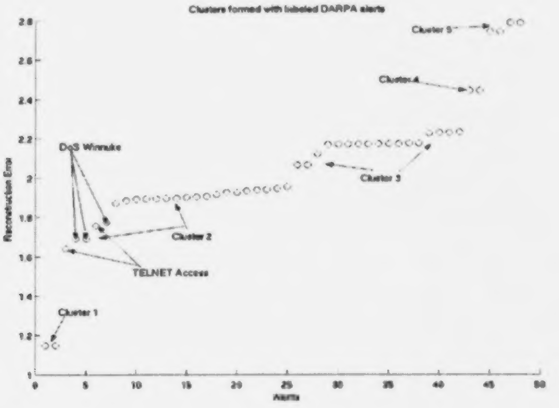


Figure C.1: Reconstruction Error Values

```
1: 1.1486: 45 46
2: 1.8686: 47 10 17 48 35 26 3 2 29 28 27 4 5 34 7 30 25 31 6 33 1 8 32
3: 2.1703: 18 11 36 12 19 13 20 14 21 15 22 23 16 37 38 39 40
4: 2.4427: 9 24
5: 2.7641: 41 43 42 44

Cluster 1: (1.1486)
--------------------
[**] [1:1104:9] WEB-MISC whisker space splice attack [**]
[Classification: Attempted Information Leak] [Priority: 2]
04/07-09:24:39.887773 206.48.44.50:2297 -> 172.16.114.50:80
TCP TTL:62 TOS:0x0 ID:5963 IpLen:20 DgmLen:41 DF
***AP*** Seq: 0xF299C093  Ack: 0x5A4D6370  Win: 0x7D78  TcpLen: 20
[Xref => http://www.wiretrip.net/rfp/pages/whitepapers/whiskerids.html]
[Xref => http://www.whitehats.com/info/IDS296]

[**] [1:1104:9] WEB-MISC whisker space splice attack [**]
[Classification: Attempted Information Leak] [Priority: 2]
04/07-09:24:39.953099 206.48.44.50:2297 -> 172.16.114.50:80
TCP TTL:62 TOS:0x0 ID:5963 IpLen:20 DgmLen:41 DF
***AP*** Seq: 0xF299C09C  Ack: 0x5A4D6370  Win: 0x7D78  TcpLen: 20
[Xref => http://www.wiretrip.net/rfp/pages/whitepapers/whiskerids.html]
[Xref => http://www.whitehats.com/info/IDS296]

Cluster 2: (1.8686)
--------------------
[**] [1:716:10] TELNET access [**]
[Classification: Not Suspicious Traffic] [Priority: 3]
04/07-17:37:12.311809 172.16.112.194:23 -> 194.7.248.153:29910
TCP TTL:64 TOS:0x10 ID:8111 IpLen:20 DgmLen:55 DF
```

```
***AP*** Seq: 0x55BB0764  Ack: 0x442FE13D  Win: 0x7FC8  TcpLen: 20
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0619]
[Xref => http://www.whitehats.com/info/IDS08]

[**] [1:1257:8] DOS Winnuke attack [**]
[Classification: Attempted Denial of Service] [Priority: 2]
04/05-11:45:27.487491 172.16.115.234:1271 -> 172.16.112.100:139
TCP TTL:127 TOS:0x0 ID:45570 IpLen:20 DgmLen:89 DF
**UAP*** Seq: 0xE20C54  Ack: 0xE37D10  Win: 0x2238  TcpLen: 20  UrgPtr: 0x31
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0153]
[Xref => http://www.securityfocus.com/bid/2010]

[**] [1:1257:8] DOS Winnuke attack [**]
[Classification: Attempted Denial of Service] [Priority: 2]
04/05-19:47:50.772413 206.48.44.18:1734 -> 172.16.115.234:139
TCP TTL:126 TOS:0x0 ID:46599 IpLen:20 DgmLen:89 DF
**UAP*** Seq: 0x29D4205  Ack: 0x29BD233  Win: 0x2238  TcpLen: 20  UrgPtr: 0x31
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0153]
[Xref => http://www.securityfocus.com/bid/2010]

[**] [1:716:10] TELNET access [**]
[Classification: Not Suspicious Traffic] [Priority: 3]
04/07-17:40:47.037327 172.16.112.100:23 -> 197.218.177.69:30373
TCP TTL:128 TOS:0x0 ID:31208 IpLen:20 DgmLen:61 DF
***AP*** Seq: 0x22998D7  Ack: 0xB2567081  Win: 0x2220  TcpLen: 20
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0619]
[Xref => http://www.whitehats.com/info/IDS08]

[**] [1:1257:8] DOS Winnuke attack [**]
[Classification: Attempted Denial of Service] [Priority: 2]
04/06-20:56:47.720344 172.16.115.234:1794 -> 172.16.112.100:139
TCP TTL:127 TOS:0x0 ID:59143 IpLen:20 DgmLen:89 DF
**UAP*** Seq: 0x2D80490  Ack: 0x2D6186F  Win: 0x2238  TcpLen: 20  UrgPtr: 0x31
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0153]
[Xref => http://www.securityfocus.com/bid/2010]

[**] [1:621:6] SCAN FIN [**]
[Classification: Attempted Information Leak] [Priority: 2]
04/05-09:43:09.166662 208.240.124.83:51945 -> 172.16.112.50:51
TCP TTL:37 TOS:0x0 ID:36064 IpLen:20 DgmLen:40
*******F Seq: 0x0  Ack: 0x0  Win: 0x1000  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS27]

[**] [1:621:6] SCAN FIN [**]
[Classification: Attempted Information Leak] [Priority: 2]
04/09-11:53:58.359515 206.186.80.111:51887 -> 172.16.113.50:7
TCP TTL:56 TOS:0x0 ID:24821 IpLen:20 DgmLen:40
*******F Seq: 0x0  Ack: 0x0  Win: 0xC00  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS27]

[**] [1:621:6] SCAN FIN [**]
[Classification: Attempted Information Leak] [Priority: 2]
04/09-11:52:12.256155 206.186.80.111:59544 -> 172.16.113.50:79
TCP TTL:48 TOS:0x0 ID:20421 IpLen:20 DgmLen:40
*******F Seq: 0x0  Ack: 0x0  Win: 0xC00  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS27]

[**] [1:621:6] SCAN FIN [**]
[Classification: Attempted Information Leak] [Priority: 2]
04/05-09:43:12.389758 208.240.124.83:56753 -> 172.16.112.50:71
TCP TTL:47 TOS:0x0 ID:26761 IpLen:20 DgmLen:40
*******F Seq: 0x0  Ack: 0x0  Win: 0x800  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS27]
```

```
[**] [1:621:6] SCAN FIN [**]
[Classification: Attempted Information Leak] [Priority: 2]
04/05-09:43:11.309436 208.240.124.83:40546 -> 172.16.112.50:45
TCP TTL:56 TOS:0x0 ID:24465 IpLen:20 DgmLen:40
*******F Seq: 0x0  Ack: 0x0  Win: 0xC00  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS27]

[**] [1:621:6] SCAN FIN [**]
[Classification: Attempted Information Leak] [Priority: 2]
04/05-09:43:10.237953 208.240.124.83:61454 -> 172.16.112.50:50
TCP TTL:48 TOS:0x0 ID:17293 IpLen:20 DgmLen:40
*******F Seq: 0x0  Ack: 0x0  Win: 0xC00  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS27]

[**] [1:621:6] SCAN FIN [**]
[Classification: Attempted Information Leak] [Priority: 2]
04/09-11:54:04.372822 206.186.80.111:51888 -> 172.16.113.50:7
TCP TTL:56 TOS:0x0 ID:47025 IpLen:20 DgmLen:40
*******F Seq: 0x0  Ack: 0x0  Win: 0xC00  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS27]

[**] [1:621:6] SCAN FIN [**]
[Classification: Attempted Information Leak] [Priority: 2]
04/09-11:55:50.468902 206.186.80.111:57112 -> 172.16.113.50:9
TCP TTL:35 TOS:0x0 ID:38447 IpLen:20 DgmLen:40
*******F Seq: 0x0  Ack: 0x0  Win: 0x800  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS27]

[**] [1:621:6] SCAN FIN [**]
[Classification: Attempted Information Leak] [Priority: 2]
04/05-09:43:28.789952 208.240.124.83:38097 -> 172.16.112.50:4
TCP TTL:42 TOS:0x0 ID:43310 IpLen:20 DgmLen:40
*******F Seq: 0x0  Ack: 0x0  Win: 0x400  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS27]

[**] [1:621:6] SCAN FIN [**]
[Classification: Attempted Information Leak] [Priority: 2]
04/09-11:57:42.595667 206.186.80.111:35145 -> 172.16.113.50:19
TCP TTL:38 TOS:0x0 ID:36231 IpLen:20 DgmLen:40
*******F Seq: 0x0  Ack: 0x0  Win: 0x400  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS27]

[**] [1:621:6] SCAN FIN [**]
[Classification: Attempted Information Leak] [Priority: 2]
04/05-09:43:13.474823 208.240.124.83:43993 -> 172.16.112.50:72
TCP TTL:38 TOS:0x0 ID:16119 IpLen:20 DgmLen:40
*******F Seq: 0x0  Ack: 0x0  Win: 0x400  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS27]

[**] [1:621:6] SCAN FIN [**]
[Classification: Attempted Information Leak] [Priority: 2]
04/05-09:43:08.073616 208.240.124.83:43170 -> 172.16.112.50:3
TCP TTL:55 TOS:0x0 ID:11515 IpLen:20 DgmLen:40
*******F Seq: 0x0  Ack: 0x0  Win: 0x800  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS27]

[**] [1:621:6] SCAN FIN [**]
[Classification: Attempted Information Leak] [Priority: 2]
04/05-09:43:14.561239 208.240.124.83:36030 -> 172.16.112.50:76
TCP TTL:51 TOS:0x0 ID:60556 IpLen:20 DgmLen:40
*******F Seq: 0x0  Ack: 0x0  Win: 0x800  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS27]

[**] [1:621:6] SCAN FIN [**]
```

```
[Classification: Attempted Information Leak] [Priority: 2]
04/09-11:55:56.489620 206.186.80.111:57113 -> 172.16.113.50:9
TCP TTL:35 TOS:0x0 ID:52543 IpLen:20 DgmLen:40
*******F Seq: 0x0  Ack: 0x0  Win: 0x800  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS27]

[**] [1:621:6] SCAN FIN [**]
[Classification: Attempted Information Leak] [Priority: 2]
04/05-09:43:21.650746 208.240.124.83:57624 -> 172.16.112.50:22
TCP TTL:36 TOS:0x0 ID:55673 IpLen:20 DgmLen:40
*******F Seq: 0x0  Ack: 0x0  Win: 0xC00  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS27]

[**] [1:621:6] SCAN FIN [**]
[Classification: Attempted Information Leak] [Priority: 2]
04/09-11:52:06.231054 206.186.80.111:59543 -> 172.16.113.50:79
TCP TTL:48 TOS:0x0 ID:55703 IpLen:20 DgmLen:40
*******F Seq: 0x0  Ack: 0x0  Win: 0xC00  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS27]

[**] [1:621:6] SCAN FIN [**]
[Classification: Attempted Information Leak] [Priority: 2]
04/09-11:57:48.616220 206.186.80.111:35146 -> 172.16.113.50:19
TCP TTL:38 TOS:0x0 ID:63498 IpLen:20 DgmLen:40
*******F Seq: 0x0  Ack: 0x0  Win: 0x400  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS27]

[**] [1:621:6] SCAN FIN [**]
[Classification: Attempted Information Leak] [Priority: 2]
04/05-09:43:15.650830 208.240.124.83:57623 -> 172.16.112.50:22
TCP TTL:36 TOS:0x0 ID:60075 IpLen:20 DgmLen:40
*******F Seq: 0x0  Ack: 0x0  Win: 0xC00  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS27]

Cluster 3: (2.1703)
--------------------
[**] [1:1257:8] DOS Winnuke attack [**]
[Classification: Attempted Denial of Service] [Priority: 2]
04/05-19:47:50.772970 206.48.44.18:1734 -> 172.16.115.234:139
TCP TTL:126 TOS:0x0 ID:46855 IpLen:20 DgmLen:236 DF
**UAP**F Seq: 0x29D4236  Ack: 0x29BD233  Win: 0x2238  TcpLen: 20  UrgPtr: 0xC4
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0153]
[Xref => http://www.securityfocus.com/bid/2010]

[**] [1:1257:8] DOS Winnuke attack [**]
[Classification: Attempted Denial of Service] [Priority: 2]
04/05-11:45:27.488041 172.16.115.234:1271 -> 172.16.112.100:139
TCP TTL:127 TOS:0x0 ID:45826 IpLen:20 DgmLen:236 DF
**UAP**F Seq: 0xE20C85  Ack: 0xE37D10  Win: 0x2238  TcpLen: 20  UrgPtr: 0xC4
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0153]
[Xref => http://www.securityfocus.com/bid/2010]

[**] [1:1257:8] DOS Winnuke attack [**]
[Classification: Attempted Denial of Service] [Priority: 2]
04/06-20:56:47.720886 172.16.115.234:1794 -> 172.16.112.100:139
TCP TTL:127 TOS:0x0 ID:59399 IpLen:20 DgmLen:236 DF
**UAP**F Seq: 0x2D804C1  Ack: 0x2D6186F  Win: 0x2238  TcpLen: 20  UrgPtr: 0xC4
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0153]
[Xref => http://www.securityfocus.com/bid/2010]

[**] [1:1257:8] DOS Winnuke attack [**]
[Classification: Attempted Denial of Service] [Priority: 2]
04/05-11:45:30.478283 172.16.115.234:1271 -> 172.16.112.100:139
TCP TTL:127 TOS:0x0 ID:46082 IpLen:20 DgmLen:285 DF
```

```
**UAP**F Seq: 0xE20C54  Ack: 0xE37D10  Win: 0x2238  TcpLen: 20  UrgPtr: 0xF5
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0153]
[Xref => http://www.securityfocus.com/bid/2010]

[**] [1:1257:8] DOS Winnuke attack [**]
[Classification: Attempted Denial of Service] [Priority: 2]
04/05-19:47:53.948406 206.48.44.18:1734 -> 172.16.115.234:139
TCP TTL:126 TOS:0x0 ID:47111 IpLen:20 DgmLen:285 DF
**UAP**F Seq: 0x29D4205  Ack: 0x29BD233  Win: 0x2238  TcpLen: 20  UrgPtr: 0xF5
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0153]
[Xref => http://www.securityfocus.com/bid/2010]

[**] [1:1257:8] DOS Winnuke attack [**]
[Classification: Attempted Denial of Service] [Priority: 2]
04/05-11:45:36.485003 172.16.115.234:1271 -> 172.16.112.100:139
TCP TTL:127 TOS:0x0 ID:46338 IpLen:20 DgmLen:285 DF
**UAP**F Seq: 0xE20C54  Ack: 0xE37D10  Win: 0x2238  TcpLen: 20  UrgPtr: 0xF5
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0153]
[Xref => http://www.securityfocus.com/bid/2010]

[**] [1:1257:8] DOS Winnuke attack [**]
[Classification: Attempted Denial of Service] [Priority: 2]
04/05-19:48:00.509550 206.48.44.18:1734 -> 172.16.115.234:139
TCP TTL:126 TOS:0x0 ID:47367 IpLen:20 DgmLen:285 DF
**UAP**F Seq: 0x29D4205  Ack: 0x29BD233  Win: 0x2238  TcpLen: 20  UrgPtr: 0xF5
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0153]
[Xref => http://www.securityfocus.com/bid/2010]

[**] [1:1257:8] DOS Winnuke attack [**]
[Classification: Attempted Denial of Service] [Priority: 2]
04/05-11:45:48.498766 172.16.115.234:1271 -> 172.16.112.100:139
TCP TTL:127 TOS:0x0 ID:46594 IpLen:20 DgmLen:285 DF
**UAP**F Seq: 0xE20C54  Ack: 0xE37D10  Win: 0x2238  TcpLen: 20  UrgPtr: 0xF5
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0153]
[Xref => http://www.securityfocus.com/bid/2010]

[**] [1:1257:8] DOS Winnuke attack [**]
[Classification: Attempted Denial of Service] [Priority: 2]
04/05-19:48:13.632019 206.48.44.18:1734 -> 172.16.115.234:139
TCP TTL:126 TOS:0x0 ID:47879 IpLen:20 DgmLen:285 DF
**UAP**F Seq: 0x29D4205  Ack: 0x29BD233  Win: 0x2238  TcpLen: 20  UrgPtr: 0xF5
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0153]
[Xref => http://www.securityfocus.com/bid/2010]

[**] [1:1257:8] DOS Winnuke attack [**]
[Classification: Attempted Denial of Service] [Priority: 2]
04/05-11:46:12.526008 172.16.115.234:1271 -> 172.16.112.100:139
TCP TTL:127 TOS:0x0 ID:47106 IpLen:20 DgmLen:285 DF
**UAP**F Seq: 0xE20C54  Ack: 0xE37D10  Win: 0x2238  TcpLen: 20  UrgPtr: 0xF5
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0153]
[Xref => http://www.securityfocus.com/bid/2010]

[**] [1:1257:8] DOS Winnuke attack [**]
[Classification: Attempted Denial of Service] [Priority: 2]
04/05-19:48:39.876731 206.48.44.18:1734 -> 172.16.115.234:139
TCP TTL:126 TOS:0x0 ID:48135 IpLen:20 DgmLen:285 DF
**UAP**F Seq: 0x29D4205  Ack: 0x29BD233  Win: 0x2238  TcpLen: 20  UrgPtr: 0xF5
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0153]
[Xref => http://www.securityfocus.com/bid/2010]

[**] [1:1257:8] DOS Winnuke attack [**]
[Classification: Attempted Denial of Service] [Priority: 2]
04/05-19:49:32.366189 206.48.44.18:1734 -> 172.16.115.234:139
TCP TTL:126 TOS:0x0 ID:48647 IpLen:20 DgmLen:285 DF
```

```
**UAP**F Seq: 0x29D4205  Ack: 0x29BD233  Win: 0x2238  TcpLen: 20  UrgPtr: 0xF5
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0153]
[Xref => http://www.securityfocus.com/bid/2010]

[**] [1:1257:8] DOS Winnuke attack [**]
[Classification: Attempted Denial of Service] [Priority: 2]
04/05-11:47:00.580526 172.16.115.234:1271 -> 172.16.112.100:139
TCP TTL:127 TOS:0x0 ID:48130 IpLen:20 DgmLen:285 DF
**UAP**F Seq: 0xE20C54  Ack: 0xE37D10  Win: 0x2238  TcpLen: 20  UrgPtr: 0xF5
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0153]
[Xref => http://www.securityfocus.com/bid/2010]

[**] [1:1257:8] DOS Winnuke attack [**]
[Classification: Attempted Denial of Service] [Priority: 2]
04/06-20:56:50.672773 172.16.115.234:1794 -> 172.16.112.100:139
TCP TTL:127 TOS:0x0 ID:59911 IpLen:20 DgmLen:285 DF
**UAP**F Seq: 0x2D80490  Ack: 0x2D6186F  Win: 0x2238  TcpLen: 20  UrgPtr: 0xF5
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0153]
[Xref => http://www.securityfocus.com/bid/2010]

[**] [1:1257:8] DOS Winnuke attack [**]
[Classification: Attempted Denial of Service] [Priority: 2]
04/06-20:56:56.679607 172.16.115.234:1794 -> 172.16.112.100:139
TCP TTL:127 TOS:0x0 ID:60167 IpLen:20 DgmLen:285 DF
**UAP**F Seq: 0x2D80490  Ack: 0x2D6186F  Win: 0x2238  TcpLen: 20  UrgPtr: 0xF5
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0153]
[Xref => http://www.securityfocus.com/bid/2010]

[**] [1:1257:8] DOS Winnuke attack [**]
[Classification: Attempted Denial of Service] [Priority: 2]
04/06-20:57:08.693240 172.16.115.234:1794 -> 172.16.112.100:139
TCP TTL:127 TOS:0x0 ID:60423 IpLen:20 DgmLen:285 DF
**UAP**F Seq: 0x2D80490  Ack: 0x2D6186F  Win: 0x2238  TcpLen: 20  UrgPtr: 0xF5
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0153]
[Xref => http://www.securityfocus.com/bid/2010]

[**] [1:1257:8] DOS Winnuke attack [**]
[Classification: Attempted Denial of Service] [Priority: 2]
04/06-20:57:32.720420 172.16.115.234:1794 -> 172.16.112.100:139
TCP TTL:127 TOS:0x0 ID:60679 IpLen:20 DgmLen:285 DF
**UAP**F Seq: 0x2D80490  Ack: 0x2D6186F  Win: 0x2238  TcpLen: 20  UrgPtr: 0xF5
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0153]
[Xref => http://www.securityfocus.com/bid/2010]

Cluster 4: (2.4427)
--------------------
[**] [1:527:8] BAD-TRAFFIC same SRC/DST [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
04/09-14:32:17.628397 172.16.113.50:25 -> 172.16.113.50:25
TCP TTL:254 TOS:0x0 ID:3868 IpLen:20 DgmLen:40
******S* Seq: 0xF1C  Ack: 0x0  Win: 0x800  TcpLen: 20
[Xref => http://www.cert.org/advisories/CA-1997-28.html]
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0016]
[Xref => http://www.securityfocus.com/bid/2666]

[**] [1:527:8] BAD-TRAFFIC same SRC/DST [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
04/05-12:48:08.463617 172.16.112.50:25 -> 172.16.112.50:25
TCP TTL:254 TOS:0x0 ID:3868 IpLen:20 DgmLen:40
******S* Seq: 0xF1C  Ack: 0x0  Win: 0x800  TcpLen: 20
[Xref => http://www.cert.org/advisories/CA-1997-28.html]
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0016]
[Xref => http://www.securityfocus.com/bid/2666]
```

```
Cluster 5: (2.7641)
--------------------
[**] [1:522:2] MISC Tiny Fragments [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
04/06-08:59:16.124877 206.48.44.50 -> 172.16.112.194
TCP TTL:62 TOS:0x0 ID:2780 IpLen:20 DgmLen:28 MF
Frag Offset: 0x0000   Frag Size: 0x0008

[**] [1:522:2] MISC Tiny Fragments [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
04/06-08:59:17.142101 206.48.44.50 -> 172.16.112.194
TCP TTL:62 TOS:0x0 ID:2781 IpLen:20 DgmLen:28 MF
Frag Offset: 0x0000   Frag Size: 0x0008

[**] [1:522:2] MISC Tiny Fragments [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
04/06-08:59:16.126994 206.48.44.50 -> 172.16.112.194
TCP TTL:62 TOS:0x0 ID:2780 IpLen:20 DgmLen:28 MF
Frag Offset: 0x0001   Frag Size: 0x0008

[**] [1:522:2] MISC Tiny Fragments [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
04/06-08:59:17.142499 206.48.44.50 -> 172.16.112.194
TCP TTL:62 TOS:0x0 ID:2781 IpLen:20 DgmLen:28 MF
Frag Offset: 0x0001   Frag Size: 0x0008
```

C.7 Incidents.org Data

The following `Incidents.org` test data was produced with 80000 epochs and an MSE of 0.06.

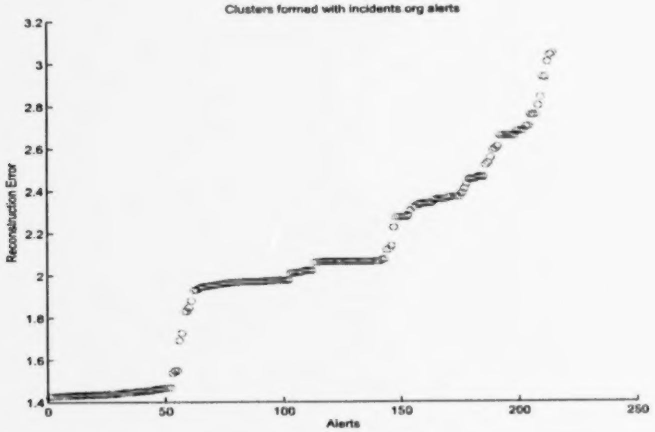


***Figure C.2:** Reconstruction Error Values*

```
Cluster 1: (1.4429)
--------------------
[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:24.096951 10.10.10.113:59195 -> 192.168.17.68:318
TCP TTL:58 TOS:0x0 ID:33580 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0xC00  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:24.412349 10.10.10.113:59194 -> 192.168.17.68:10
TCP TTL:55 TOS:0x0 ID:33213 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x1000  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:23.791126 10.10.10.113:59194 -> 192.168.17.68:446
TCP TTL:55 TOS:0x0 ID:36902 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x1000  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:23.488595 10.10.10.113:59195 -> 192.168.17.68:415
TCP TTL:46 TOS:0x0 ID:30853 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0xC00  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]
```

```
[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:23.790722 10.10.10.113:59195 -> 192.168.17.68:806
TCP TTL:47 TOS:0x0 ID:36012 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x1000  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:23.791229 10.10.10.113:59194 -> 192.168.17.68:549
TCP TTL:52 TOS:0x0 ID:40144 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x400  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:23.791329 10.10.10.113:59194 -> 192.168.17.68:706
TCP TTL:50 TOS:0x0 ID:40842 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0xC00  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:23.488548 10.10.10.113:59195 -> 192.168.17.68:1015
TCP TTL:53 TOS:0x0 ID:23833 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x800  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:23.488551 10.10.10.113:59195 -> 192.168.17.68:896
TCP TTL:47 TOS:0x0 ID:25885 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x1000  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:23.130812 10.10.10.113:59194 -> 192.168.17.68:904
TCP TTL:47 TOS:0x0 ID:41650 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x1000  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:24.097120 10.10.10.113:59195 -> 192.168.17.68:446
TCP TTL:54 TOS:0x0 ID:44626 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0xC00  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:23.488289 10.10.10.113:59195 -> 192.168.17.68:904
TCP TTL:52 TOS:0x0 ID:22463 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x400  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:24.097274 10.10.10.113:59195 -> 192.168.17.68:5490
TCP TTL:52 TOS:0x0 ID:25873 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x400  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]
```

```
[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:59:18.500725 10.10.10.113:59194 -> 192.168.17.68:651
TCP TTL:49 TOS:0x0 ID:43132 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x800  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:23.488231 10.10.10.113:59195 -> 192.168.17.68:300
TCP TTL:52 TOS:0x0 ID:21228 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x400  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:23.131117 10.10.10.113:59194 -> 192.168.17.68:415
TCP TTL:37 TOS:0x0 ID:38308 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x800  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:23.130864 10.10.10.113:59194 -> 192.168.17.68:3462
TCP TTL:44 TOS:0x0 ID:24121 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x400  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:24.097375 10.10.10.113:59194 -> 192.168.17.68:3457
TCP TTL:42 TOS:0x0 ID:40601 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0xC00  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:23.790793 10.10.10.113:59194 -> 192.168.17.68:1390
TCP TTL:40 TOS:0x0 ID:24566 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x400  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:24.412198 10.10.10.113:59194 -> 192.168.17.68:1467
TCP TTL:42 TOS:0x0 ID:43345 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0xC00  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:23.488542 10.10.10.113:59195 -> 192.168.17.68:3462
TCP TTL:37 TOS:0x0 ID:26392 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x800  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:23.488546 10.10.10.113:59195 -> 192.168.17.68:306
TCP TTL:41 TOS:0x0 ID:21576 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x800  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
```

```
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:24.097173 10.10.10.113:59195 -> 192.168.17.68:203
TCP TTL:44 TOS:0x0 ID:20553 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x400  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:23.131067 10.10.10.113:59194 -> 192.168.17.68:4133
TCP TTL:48 TOS:0x0 ID:45399 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x400  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:23.791279 10.10.10.113:59194 -> 192.168.17.68:5490
TCP TTL:53 TOS:0x0 ID:19159 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x800  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:24.097003 10.10.10.113:59195 -> 192.168.17.68:249
TCP TTL:42 TOS:0x0 ID:47049 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0xC00  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:24.412044 10.10.10.113:59194 -> 192.168.17.68:497
TCP TTL:56 TOS:0x0 ID:16002 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x400  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:23.488554 10.10.10.113:59195 -> 192.168.17.68:4133
TCP TTL:43 TOS:0x0 ID:19661 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x1000  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:23.130914 10.10.10.113:59194 -> 192.168.17.68:306
TCP TTL:39 TOS:0x0 ID:47965 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x1000  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:24.097223 10.10.10.113:59195 -> 192.168.17.68:549
TCP TTL:40 TOS:0x0 ID:47975 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x400  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:23.130710 10.10.10.113:59194 -> 192.168.17.68:443
TCP TTL:45 TOS:0x0 ID:50637 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x800  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
```

```
11/18-13:57:23.130710 10.10.10.113:59194 -> 192.168.17.68:443
TCP TTL:45 TOS:0x0 ID:50637 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x800  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:24.412299 10.10.10.113:59194 -> 192.168.17.68:361
TCP TTL:54 TOS:0x0 ID:54937 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0xC00  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:23.130762 10.10.10.113:59194 -> 192.168.17.68:300
TCP TTL:59 TOS:0x0 ID:11324 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x1000  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:23.130762 10.10.10.113:59194 -> 192.168.17.68:300
TCP TTL:59 TOS:0x0 ID:11324 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x1000  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:24.097324 10.10.10.113:59195 -> 192.168.17.68:706
TCP TTL:57 TOS:0x0 ID:10850 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x800  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:24.096882 10.10.10.113:59195 -> 192.168.17.68:1390
TCP TTL:58 TOS:0x0 ID:57453 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0xC00  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:23.130964 10.10.10.113:59194 -> 192.168.17.68:1015
TCP TTL:56 TOS:0x0 ID:57344 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x400  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:24.411973 10.10.10.113:59195 -> 192.168.17.68:3457
TCP TTL:59 TOS:0x0 ID:8985 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x1000  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:25.339047 10.10.10.113:59195 -> 192.168.17.68:7273
TCP TTL:48 TOS:0x0 ID:54546 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x400  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:23.790946 10.10.10.113:59194 -> 192.168.17.68:1418
```

```
TCP TTL:53 TOS:0x0 ID:57918 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x800  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:23.488646 10.10.10.113:59194 -> 192.168.17.68:806
TCP TTL:49 TOS:0x0 ID:8472 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x800  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:24.412247 10.10.10.113:59194 -> 192.168.17.68:459
TCP TTL:39 TOS:0x0 ID:9660 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x1000  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:23.131015 10.10.10.113:59194 -> 192.168.17.68:896
TCP TTL:43 TOS:0x0 ID:8862 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x1000  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:24.097055 10.10.10.113:59195 -> 192.168.17.68:1418
TCP TTL:53 TOS:0x0 ID:60948 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x800  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:23.791179 10.10.10.113:59194 -> 192.168.17.68:203
TCP TTL:51 TOS:0x0 ID:5063 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x1000  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:23.790845 10.10.10.113:59194 -> 192.168.17.68:318
TCP TTL:54 TOS:0x0 ID:62352 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0xC00  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:23.487553 10.10.10.113:59195 -> 192.168.17.68:1027
TCP TTL:58 TOS:0x0 ID:63297 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0xC00  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:23.790895 10.10.10.113:59194 -> 192.168.17.68:249
TCP TTL:46 TOS:0x0 ID:3540 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0xC00  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:24.412147 10.10.10.113:59194 -> 192.168.17.68:1531
TCP TTL:52 TOS:0x0 ID:1989 IpLen:20 DgmLen:40
```

```
******** Seq: 0x0  Ack: 0x0  Win: 0x400  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:25.339099 10.10.10.113:59194 -> 192.168.17.68:3985
TCP TTL:42 TOS:0x0 ID:62540 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0xC00  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:23.130647 10.10.10.113:59194 -> 192.168.17.68:1027
TCP TTL:51 TOS:0x0 ID:1556 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x1000  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

Cluster 2: (1.5456)
--------------------
[**] [1:623:2] SCAN NULL [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:24.412096 10.10.10.113:59194 -> 192.168.17.68:32771
TCP TTL:37 TOS:0x0 ID:46742 IpLen:20 DgmLen:40
******** Seq: 0x0  Ack: 0x0  Win: 0x800  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS4]

[**] [116:97:1] (snort_decoder): Short UDP packet, length field > payload length [**]
[Classification: Potential Corporate Privacy Violation] [Priority: 1]
10/29-14:16:27.826507 211.223.8.214:0 -> 32.245.161.79:0
UDP TTL:109 TOS:0x0 ID:25767 IpLen:20 DgmLen:78
Len: 129

[**] [116:97:1] (snort_decoder): Short UDP packet, length field > payload length [**]
[Classification: Potential Corporate Privacy Violation] [Priority: 1]
10/27-21:10:12.336507 80.63.124.198:0 -> 32.245.98.91:0
UDP TTL:112 TOS:0x0 ID:314 IpLen:20 DgmLen:78
Len: 129

Cluster 3: (1.6921)
--------------------
[**] [1:615:5] SCAN SOCKS Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/17-06:57:26.936507 202.108.254.200:51622 -> 170.129.107.88:1080
TCP TTL:45 TOS:0x0 ID:30451 IpLen:20 DgmLen:40
******S* Seq: 0x63DF049B  Ack: 0x63DF049B  Win: 0x400  TcpLen: 20
[Xref => http://help.undernet.org/proxyscan/]

Cluster 4: (1.7271)
--------------------
[**] [1:618:5] SCAN Squid Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/17-06:57:23.666507 202.108.254.200:26931 -> 170.129.107.88:3128
TCP TTL:45 TOS:0x0 ID:42535 IpLen:20 DgmLen:40
******S* Seq: 0x23128300  Ack: 0x23128300  Win: 0x400  TcpLen: 20

Cluster 5: (1.8305)
--------------------
[**] [1:618:5] SCAN Squid Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-16:06:21.366507 206.48.61.139:4006 -> 170.129.23.239:3128
TCP TTL:114 TOS:0x0 ID:24710 IpLen:20 DgmLen:48 DF
******S* Seq: 0x13D84F7  Ack: 0x0  Win: 0x2000  TcpLen: 28
TCP Options (4) => MSS: 536 NOP NOP SackOK
```

```
[**] [1:618:5] SCAN Squid Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-16:06:24.316507 206.48.61.139:4006 -> 170.129.23.239:3128
TCP TTL:114 TOS:0x0 ID:26758 IpLen:20 DgmLen:48 DF
******S* Seq: 0x13D84F7  Ack: 0x0  Win: 0x2000  TcpLen: 28
TCP Options (4) => MSS: 536 NOP NOP SackOK

Cluster 6: (1.8487)
--------------------
[**] [116:97:1] (snort_decoder): Short UDP packet, length field > payload length [**]
[Classification: Potential Corporate Privacy Violation] [Priority: 1]
10/24-07:16:45.936507 80.78.226.31:0 -> 32.245.132.116:0
UDP TTL:18 TOS:0x0 ID:3631 IpLen:20 DgmLen:513
Len: 63251

Cluster 7: (1.8803)
--------------------
[**] [116:97:1] (snort_decoder): Short UDP packet, length field > payload length [**]
11/18-13:57:31.994673 10.10.10.2:0 -> 10.10.10.214:0
UDP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:132 DF
Len: 104

Cluster 8: (1.9643)
--------------------
[**] [1:504:4] MISC source port 53 to <1024 [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
11/15-11:24:29.636507 206.102.126.101:53 -> 170.129.50.122:53
TCP TTL:54 TOS:0x0 ID:24271 IpLen:20 DgmLen:40
******S* Seq: 0x4E80C2A7  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS07]

[**] [116:97:1] (snort_decoder): Short UDP packet, length field > payload length [**]
11/18-13:57:31.994857 10.10.10.214:0 -> 10.10.10.2:0
UDP TTL:64 TOS:0x0 ID:36711 IpLen:20 DgmLen:86 DF
Len: 58

[**] [1:620:6] SCAN Proxy Port 8080 attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/17-06:57:31.896507 202.108.254.200:8576 -> 170.129.38.209:8080
TCP TTL:46 TOS:0x0 ID:61101 IpLen:20 DgmLen:40
******S* Seq: 0x538A1F3F  Ack: 0x538A1F3F  Win: 0x400  TcpLen: 20

[**] [1:474:1] ICMP superscan echo [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:27.548777 10.10.10.164 -> 172.22.201.34
ICMP TTL:128 TOS:0x0 ID:31890 IpLen:20 DgmLen:36
Type:8  Code:0  ID:1024   Seq:41217  ECHO

[**] [1:474:1] ICMP superscan echo [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:27.556061 10.10.10.164 -> 172.22.201.35
ICMP TTL:128 TOS:0x0 ID:31892 IpLen:20 DgmLen:36
Type:8  Code:0  ID:1024   Seq:41473  ECHO

[**] [1:474:1] ICMP superscan echo [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:27.565776 10.10.10.164 -> 172.22.201.36
ICMP TTL:128 TOS:0x0 ID:31894 IpLen:20 DgmLen:36
Type:8  Code:0  ID:1024   Seq:41729  ECHO

[**] [1:474:1] ICMP superscan echo [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:27.575564 10.10.10.164 -> 172.22.201.37
ICMP TTL:128 TOS:0x0 ID:31896 IpLen:20 DgmLen:36
```

```
Type:8  Code:0  ID:1024   Seq:41985  ECHO

[**] [1:474:1] ICMP superscan echo [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:27.585344 10.10.10.164 -> 172.22.201.38
ICMP TTL:128 TOS:0x0 ID:31898 IpLen:20 DgmLen:36
Type:8  Code:0  ID:1024   Seq:42241  ECHO

[**] [1:474:1] ICMP superscan echo [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:27.596488 10.10.10.164 -> 172.22.201.39
ICMP TTL:128 TOS:0x0 ID:31900 IpLen:20 DgmLen:36
Type:8  Code:0  ID:1024   Seq:42497  ECHO

[**] [1:474:1] ICMP superscan echo [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:27.608241 10.10.10.164 -> 172.22.201.40
ICMP TTL:128 TOS:0x0 ID:31902 IpLen:20 DgmLen:36
Type:8  Code:0  ID:1024   Seq:42753  ECHO

[**] [1:474:1] ICMP superscan echo [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:27.615723 10.10.10.164 -> 172.22.201.41
ICMP TTL:128 TOS:0x0 ID:31904 IpLen:20 DgmLen:36
Type:8  Code:0  ID:1024   Seq:43009  ECHO

[**] [1:474:1] ICMP superscan echo [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:27.625682 10.10.10.164 -> 172.22.201.42
ICMP TTL:128 TOS:0x0 ID:31906 IpLen:20 DgmLen:36
Type:8  Code:0  ID:1024   Seq:43265  ECHO

[**] [1:474:1] ICMP superscan echo [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:27.635621 10.10.10.164 -> 172.22.201.43
ICMP TTL:128 TOS:0x0 ID:31908 IpLen:20 DgmLen:36
Type:8  Code:0  ID:1024   Seq:43521  ECHO

[**] [1:474:1] ICMP superscan echo [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:27.646350 10.10.10.164 -> 172.22.201.44
ICMP TTL:128 TOS:0x0 ID:31910 IpLen:20 DgmLen:36
Type:8  Code:0  ID:1024   Seq:43777  ECHO

[**] [1:474:1] ICMP superscan echo [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:27.655651 10.10.10.164 -> 172.22.201.45
ICMP TTL:128 TOS:0x0 ID:31912 IpLen:20 DgmLen:36
Type:8  Code:0  ID:1024   Seq:44033  ECHO

[**] [1:628:3] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-12:02:45.976507 167.79.91.3:80 -> 170.129.50.122:53
TCP TTL:49 TOS:0x0 ID:11661 IpLen:20 DgmLen:40
***A**** Seq: 0x2A5  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]

[**] [1:628:3] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-12:02:46.046507 167.79.91.3:80 -> 170.129.50.122:53
TCP TTL:47 TOS:0x0 ID:11664 IpLen:20 DgmLen:40
***A**** Seq: 0x2A7  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]
```

```
[**] [1:474:1] ICMP superscan echo [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:27.667713 10.10.10.164 -> 172.22.201.46
ICMP TTL:128 TOS:0x0 ID:31914 IpLen:20 DgmLen:36
Type:8  Code:0  ID:1024   Seq:44289  ECHO

[**] [1:628:3] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/17-05:50:51.616507 61.218.161.210:80 -> 170.129.44.252:80
TCP TTL:47 TOS:0x0 ID:7308 IpLen:20 DgmLen:40
***A**** Seq: 0x229  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]

[**] [1:474:1] ICMP superscan echo [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:27.675635 10.10.10.164 -> 172.22.201.47
ICMP TTL:128 TOS:0x0 ID:31916 IpLen:20 DgmLen:36
Type:8  Code:0  ID:1024   Seq:44545  ECHO

[**] [1:628:3] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/17-05:50:59.856507 163.23.238.9:80 -> 170.129.44.252:80
TCP TTL:43 TOS:0x0 ID:7419 IpLen:20 DgmLen:40
***A**** Seq: 0x250  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]

[**] [1:474:1] ICMP superscan echo [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:27.685685 10.10.10.164 -> 172.22.201.48
ICMP TTL:128 TOS:0x0 ID:31918 IpLen:20 DgmLen:36
Type:8  Code:0  ID:1024   Seq:44801  ECHO

[**] [1:628:3] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-14:28:54.256507 202.29.28.1:80 -> 170.129.238.112:80
TCP TTL:45 TOS:0x0 ID:23204 IpLen:20 DgmLen:40
***A**** Seq: 0x2E6  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]

[**] [1:628:3] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-14:28:54.256507 202.29.28.1:80 -> 170.129.238.112:80
TCP TTL:45 TOS:0x0 ID:23204 IpLen:20 DgmLen:40
***A**** Seq: 0x2E6  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]

[**] [1:628:3] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-14:29:04.266507 202.29.28.1:80 -> 170.129.238.112:80
TCP TTL:45 TOS:0x0 ID:23404 IpLen:20 DgmLen:40
***A**** Seq: 0x348  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]

[**] [1:628:3] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-14:29:14.306507 202.29.28.1:80 -> 170.129.238.112:80
TCP TTL:45 TOS:0x0 ID:23584 IpLen:20 DgmLen:40
***A**** Seq: 0x3A6  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]

[**] [1:628:3] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-14:29:24.186507 202.29.28.1:80 -> 170.129.238.112:80
TCP TTL:45 TOS:0x0 ID:23770 IpLen:20 DgmLen:40
```

```
***A**** Seq: 0x0  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]

[**] [1:628:3] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-14:29:24.186507 202.29.28.1:80 -> 170.129.238.112:80
TCP TTL:45 TOS:0x0 ID:23770 IpLen:20 DgmLen:40
***A**** Seq: 0x0  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]

[**] [1:628:3] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-14:29:34.136507 202.29.28.1:80 -> 170.129.238.112:80
TCP TTL:45 TOS:0x0 ID:23947 IpLen:20 DgmLen:40
***A**** Seq: 0x5D  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]

[**] [1:474:1] ICMP superscan echo [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:27.696617 10.10.10.164 -> 172.22.201.49
ICMP TTL:128 TOS:0x0 ID:31920 IpLen:20 DgmLen:36
Type:8  Code:0  ID:1024   Seq:45057  ECHO

[**] [1:628:3] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-14:50:33.266507 61.218.161.210:80 -> 170.129.151.28:80
TCP TTL:48 TOS:0x0 ID:32 IpLen:20 DgmLen:40
***A**** Seq: 0x17A  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]

[**] [1:474:1] ICMP superscan echo [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:27.706041 10.10.10.164 -> 172.22.201.50
ICMP TTL:128 TOS:0x0 ID:31922 IpLen:20 DgmLen:36
Type:8  Code:0  ID:1024   Seq:45313  ECHO

[**] [1:628:3] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-12:51:49.906507 61.218.161.202:80 -> 170.129.14.62:80
TCP TTL:48 TOS:0x0 ID:28299 IpLen:20 DgmLen:40
***A**** Seq: 0x11B  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]

[**] [1:628:3] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-14:50:38.206507 163.23.238.9:80 -> 170.129.151.28:80
TCP TTL:44 TOS:0x0 ID:336 IpLen:20 DgmLen:40
***A**** Seq: 0x1EC  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]

[**] [1:628:3] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-10:10:03.816507 61.218.161.202:80 -> 170.129.19.170:80
TCP TTL:48 TOS:0x0 ID:30084 IpLen:20 DgmLen:40
***A**** Seq: 0x134  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]

[**] [1:628:3] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-10:10:08.786507 61.218.161.202:80 -> 170.129.19.170:80
TCP TTL:48 TOS:0x0 ID:30366 IpLen:20 DgmLen:40
***A**** Seq: 0x198  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]
```

```
[**] [1:628:3] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-10:10:13.826507 61.218.161.210:80 -> 170.129.19.170:80
TCP TTL:48 TOS:0x0 ID:30662 IpLen:20 DgmLen:40
***A**** Seq: 0x20A  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]

[**] [1:628:3] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-10:10:13.826507 61.218.161.210:80 -> 170.129.19.170:80
TCP TTL:48 TOS:0x0 ID:30662 IpLen:20 DgmLen:40
***A**** Seq: 0x20A  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]

[**] [1:628:3] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-10:10:18.856507 61.218.161.210:80 -> 170.129.19.170:80
TCP TTL:48 TOS:0x0 ID:30943 IpLen:20 DgmLen:40
***A**** Seq: 0x278  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]

[**] [1:628:3] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-10:10:18.856507 61.218.161.210:80 -> 170.129.19.170:80
TCP TTL:48 TOS:0x0 ID:30943 IpLen:20 DgmLen:40
***A**** Seq: 0x278  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]

[**] [1:628:3] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-10:10:23.856507 163.23.238.9:80 -> 170.129.19.170:80
TCP TTL:44 TOS:0x0 ID:31290 IpLen:20 DgmLen:40
***A**** Seq: 0x300  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]

Cluster 9: (2.0181)
--------------------
[**] [1:628:3] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-12:14:11.266507 61.222.14.98:80 -> 170.129.81.112:80
TCP TTL:49 TOS:0x0 ID:48192 IpLen:20 DgmLen:40
***A**** Seq: 0x144  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]

[**] [1:628:3] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-12:14:16.266507 61.222.14.98:80 -> 170.129.81.112:80
TCP TTL:49 TOS:0x0 ID:48734 IpLen:20 DgmLen:40
***A**** Seq: 0x1A8  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]

[**] [1:628:3] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-12:14:21.266507 61.222.192.98:80 -> 170.129.81.112:80
TCP TTL:49 TOS:0x0 ID:49258 IpLen:20 DgmLen:40
***A**** Seq: 0x20A  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]

[**] [1:628:3] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-12:14:26.276507 61.222.192.98:80 -> 170.129.81.112:80
TCP TTL:49 TOS:0x0 ID:49820 IpLen:20 DgmLen:40
***A**** Seq: 0x271  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]
```

```
[**] [1:628:3] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-12:14:26.276507 61.222.192.98:80 -> 170.129.81.112:80
TCP TTL:49 TOS:0x0 ID:49820 IpLen:20 DgmLen:40
***A**** Seq: 0x271  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]

[**] [1:628:3] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-12:14:41.366507 210.66.117.5:80 -> 170.129.81.112:80
TCP TTL:47 TOS:0x0 ID:51450 IpLen:20 DgmLen:40
***A**** Seq: 0x39E  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]

[**] [1:628:3] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-12:14:46.366507 210.66.117.5:80 -> 170.129.81.112:80
TCP TTL:47 TOS:0x0 ID:51968 IpLen:20 DgmLen:40
***A**** Seq: 0x400  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]

[**] [1:465:1] ICMP ISS Pinger [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:59:08.853509 10.10.10.165 -> 172.20.201.221
ICMP TTL:128 TOS:0x0 ID:3388 IpLen:20 DgmLen:44
Type:8  Code:0  ID:34779   Seq:8960  ECHO
[Xref => http://www.whitehats.com/info/IDS158]

[**] [1:465:1] ICMP ISS Pinger [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:59:08.853564 10.10.10.165 -> 172.20.201.49
ICMP TTL:128 TOS:0x0 ID:3389 IpLen:20 DgmLen:44
Type:8  Code:0  ID:34779   Seq:8960  ECHO
[Xref => http://www.whitehats.com/info/IDS158]

[**] [1:465:1] ICMP ISS Pinger [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:59:08.853618 10.10.10.165 -> 172.20.201.158
ICMP TTL:128 TOS:0x0 ID:3390 IpLen:20 DgmLen:44
Type:8  Code:0  ID:34779   Seq:8960  ECHO
[Xref => http://www.whitehats.com/info/IDS158]

Cluster 10: (2.0663)
--------------------
[**] [1:628:3] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-14:50:18.166507 61.218.161.202:80 -> 170.129.151.28:80
TCP TTL:48 TOS:0x0 ID:64698 IpLen:20 DgmLen:40
***A**** Seq: 0x26  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]

[**] [1:628:3] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-14:50:23.136507 61.218.161.202:80 -> 170.129.151.28:80
TCP TTL:48 TOS:0x0 ID:64986 IpLen:20 DgmLen:40
***A**** Seq: 0x8C  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]

[**] [1:628:3] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-14:50:28.126507 61.218.161.210:80 -> 170.129.151.28:80
TCP TTL:48 TOS:0x0 ID:65271 IpLen:20 DgmLen:40
***A**** Seq: 0x100  Ack: 0x0  Win: 0x578  TcpLen: 20
```

```
[Xref => http://www.whitehats.com/info/IDS28]

[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/14-09:29:14.826507 255.255.255.255:31337 -> 170.129.172.186:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]

[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/14-09:32:53.016507 255.255.255.255:31337 -> 170.129.132.79:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]

[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/14-09:49:26.156507 255.255.255.255:31337 -> 170.129.129.188:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]

[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/14-14:04:08.966507 255.255.255.255:31337 -> 170.129.190.188:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]

[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/14-21:29:40.696507 255.255.255.255:31337 -> 170.129.161.211:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]

[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/14-09:29:14.826507 255.255.255.255:31337 -> 170.129.172.186:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]

[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/14-09:32:53.016507 255.255.255.255:31337 -> 170.129.132.79:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]

[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/14-09:49:26.156507 255.255.255.255:31337 -> 170.129.129.188:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]

[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/14-10:10:22.596507 255.255.255.255:31337 -> 170.129.195.178:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]
```

```
[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/14-10:44:04.836507 255.255.255.255:31337 -> 170.129.30.34:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]

[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/14-11:44:56.156507 255.255.255.255:31337 -> 170.129.137.174:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]

[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/14-14:01:20.986507 255.255.255.255:31337 -> 170.129.23.133:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]

[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/14-14:04:08.966507 255.255.255.255:31337 -> 170.129.190.188:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]

[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/14-16:14:32.966507 255.255.255.255:31337 -> 170.129.192.22:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]

[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/14-16:16:56.986507 255.255.255.255:31337 -> 170.129.134.5:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]

[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/14-16:17:12.036507 255.255.255.255:31337 -> 170.129.156.132:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]

[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/14-16:30:57.086507 255.255.255.255:31337 -> 170.129.146.62:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]

[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/14-16:47:18.156507 255.255.255.255:31337 -> 170.129.176.42:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]
```

```
[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/14-17:06:30.606507 255.255.255.255:31337 -> 170.129.80.5:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]

[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/14-17:11:03.616507 255.255.255.255:31337 -> 170.129.1.102:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]

[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/14-17:23:24.646507 255.255.255.255:31337 -> 170.129.41.171:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]

[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/14-19:34:10.596507 255.255.255.255:31337 -> 170.129.94.129:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]

[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/14-19:38:10.756507 255.255.255.255:31337 -> 170.129.181.145:515
TCP TTL:15 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]

[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/16-05:44:56.686507 255.255.255.255:31337 -> 170.128.23.189:515
TCP TTL:14 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]

[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/16-06:16:56.956507 255.255.255.255:31337 -> 170.129.207.122:515
TCP TTL:14 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]

[**] [1:184:4] BACKDOOR Q access [**]
[Classification: Misc activity] [Priority: 3]
11/14-13:35:08.896507 255.255.255.255:31337 -> 170.129.200.84:515
TCP TTL:14 TOS:0x0 ID:0 IpLen:20 DgmLen:43
***A*R** Seq: 0x0  Ack: 0x0  Win: 0x0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS203]

[**] [1:474:1] ICMP superscan echo [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:29.188265 10.10.10.164 -> 172.22.201.98
ICMP TTL:128 TOS:0x0 ID:31976 IpLen:20 DgmLen:36
Type:8  Code:0  ID:1024   Seq:57601  ECHO

[**] [1:474:1] ICMP superscan echo [**]
[Classification: Attempted Information Leak] [Priority: 2]
```

```
11/18-13:57:29.206515 10.10.10.164 -> 172.22.201.99
ICMP TTL:128 TOS:0x0 ID:31977 IpLen:20 DgmLen:36
Type:8  Code:0  ID:1024   Seq:57857  ECHO

Cluster 11: (2.1219)
--------------------
[**] [116:46:1] (snort_decoder) WARNING: TCP Data Offset is less than 5! [**]
11/15-23:11:54.416507 68.41.28.138:0 -> 170.129.23.60:0
TCP TTL:106 TOS:0x0 ID:39864 IpLen:20 DgmLen:48 DF
******** Seq: 0x5B4F202C  Ack: 0x0  Win: 0x7002  TcpLen: 0

[**] [1:615:5] SCAN SOCKS Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:23.868530 10.10.10.231:4931 -> 192.168.17.9:1080
TCP TTL:128 TOS:0x0 ID:52948 IpLen:20 DgmLen:48 DF
******S* Seq: 0x8F4E9FE9  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1360 NOP NOP SackOK
[Xref => http://help.undernet.org/proxyscan/]

Cluster 12: (2.1392)
--------------------
[**] [116:46:1] (snort_decoder) WARNING: TCP Data Offset is less than 5! [**]
[Classification: Potential Corporate Privacy Violation] [Priority: 1]
10/30-21:25:03.456507 218.44.144.208:0 -> 207.166.87.40:0
TCP TTL:105 TOS:0x0 ID:6615 IpLen:20 DgmLen:40 DF
******** Seq: 0xA42E4500  Ack: 0x5FA130D  Win: 0x4223  TcpLen: 0

Cluster 13: (2.2267)
--------------------
[**] [116:46:1] (snort_decoder) WARNING: TCP Data Offset is less than 5! [**]
[Classification: Potential Corporate Privacy Violation] [Priority: 1]
09/30-08:05:08.496507 62.121.131.17:0 -> 115.74.197.194:0
TCP TTL:45 TOS:0x0 ID:17767 IpLen:20 DgmLen:44
******S* Seq: 0xFE6BF4A  Ack: 0x10713BF7  Win: 0xFFFF  TcpLen: 0

Cluster 14: (2.2759)
--------------------
[**] [1:527:4] BAD-TRAFFIC same SRC/DST [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
11/16-03:26:16.456507 170.129.71.37 -> 170.129.71.37
IGMP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:28
[Xref => http://www.cert.org/advisories/CA-1997-28.html][Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-

[**] [1:527:4] BAD-TRAFFIC same SRC/DST [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
11/16-03:26:16.456507 170.129.71.74 -> 170.129.71.74
IGMP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:28
[Xref => http://www.cert.org/advisories/CA-1997-28.html][Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-

[**] [1:527:4] BAD-TRAFFIC same SRC/DST [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
11/16-03:26:16.456507 170.129.71.7 -> 170.129.71.7
IGMP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:28
[Xref => http://www.cert.org/advisories/CA-1997-28.html][Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-

[**] [1:527:4] BAD-TRAFFIC same SRC/DST [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
11/16-03:26:16.456507 170.129.71.26 -> 170.129.71.26
IGMP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:28
[Xref => http://www.cert.org/advisories/CA-1997-28.html][Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-

[**] [1:527:4] BAD-TRAFFIC same SRC/DST [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
11/16-03:26:16.456507 170.129.71.31 -> 170.129.71.31
```

```
IGMP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:28
[Xref => http://www.cert.org/advisories/CA-1997-28.html][Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-

[**] [1:628:3] SCAN nmap TCP [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-10:52:22.116507 63.211.17.228:80 -> 170.129.50.120:63874
TCP TTL:54 TOS:0x0 ID:563 IpLen:20 DgmLen:40
***A**** Seq: 0x3DC  Ack: 0x0  Win: 0x578  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS28]

Cluster 15: (2.3036)
--------------------
[**] [1:524:7] BAD-TRAFFIC tcp port 0 traffic [**]
[Classification: Misc activity] [Priority: 3]
11/15-11:56:39.336507 211.47.255.24:42742 -> 170.129.21.249:0
TCP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0xAECA8CC2  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

Cluster 16: (2.3516)
--------------------
[**] [116:46:1] (snort_decoder) WARNING: TCP Data Offset is less than 5! [**]
[Classification: Potential Corporate Privacy Violation] [Priority: 1]
11/12-03:07:49.886507 210.243.145.141:0 -> 207.166.159.139:0
TCP TTL:237 TOS:0x0 ID:0 IpLen:20 DgmLen:40
*****R** Seq: 0x158662C0  Ack: 0x158662C0  Win: 0x0  TcpLen: 16

[**] [116:46:1] (snort_decoder) WARNING: TCP Data Offset is less than 5! [**]
[Classification: Potential Corporate Privacy Violation] [Priority: 1]
11/12-03:07:49.886507 210.243.145.141:0 -> 207.166.159.139:0
TCP TTL:237 TOS:0x0 ID:0 IpLen:20 DgmLen:40
*****R** Seq: 0x158662C0  Ack: 0x158662C0  Win: 0x0  TcpLen: 16

[**] [116:46:1] (snort_decoder) WARNING: TCP Data Offset is less than 5! [**]
[Classification: Potential Corporate Privacy Violation] [Priority: 1]
11/10-05:38:56.936507 62.13.27.29:0 -> 207.166.25.86:0
TCP TTL:234 TOS:0x0 ID:0 IpLen:20 DgmLen:40
*****R** Seq: 0xA02C678  Ack: 0xA02C678  Win: 0x0  TcpLen: 8

[**] [1:618:5] SCAN Squid Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-16:00:56.996507 66.159.18.66:43518 -> 170.129.50.120:3128
TCP TTL:53 TOS:0x0 ID:50174 IpLen:20 DgmLen:60 DF
******S* Seq: 0xBF3AC0C  Ack: 0x0  Win: 0x16D0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 48656370 0 NOP WS: 0

[**] [1:618:5] SCAN Squid Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-16:00:56.996507 66.159.18.66:43518 -> 170.129.50.120:3128
TCP TTL:53 TOS:0x0 ID:50174 IpLen:20 DgmLen:60 DF
******S* Seq: 0xBF3AC0C  Ack: 0x0  Win: 0x16D0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 48656370 0 NOP WS: 0

[**] [1:556:5] P2P Outbound GNUTella client request [**]
[Classification: Potential Corporate Privacy Violation] [Priority: 1]
11/14-15:43:37.096507 170.129.50.120:61121 -> 24.65.114.32:6003
TCP TTL:123 TOS:0x0 ID:22720 IpLen:20 DgmLen:158 DF
***AP*** Seq: 0x5A0CA92C  Ack: 0x53A65413  Win: 0x4038  TcpLen: 20

[**] [1:556:5] P2P Outbound GNUTella client request [**]
[Classification: Potential Corporate Privacy Violation] [Priority: 1]
11/14-15:43:37.096507 170.129.50.120:61121 -> 24.65.114.32:6003
TCP TTL:123 TOS:0x0 ID:22720 IpLen:20 DgmLen:158 DF
***AP*** Seq: 0x5A0CA92C  Ack: 0x53A65413  Win: 0x4038  TcpLen: 20
```

```
[**] [1:556:5] P2P Outbound GNUTella client request [**]
[Classification: Potential Corporate Privacy Violation] [Priority: 1]
11/14-15:43:37.316507 170.129.50.120:61122 -> 24.65.114.32:6003
TCP TTL:123 TOS:0x0 ID:22766 IpLen:20 DgmLen:62 DF
***AP*** Seq: 0x5A129557  Ack: 0x53A7A3BA  Win: 0x4038  TcpLen: 20

[**] [1:556:5] P2P Outbound GNUTella client request [**]
[Classification: Potential Corporate Privacy Violation] [Priority: 1]
11/14-15:43:37.316507 170.129.50.120:61122 -> 24.65.114.32:6003
TCP TTL:123 TOS:0x0 ID:22766 IpLen:20 DgmLen:62 DF
***AP*** Seq: 0x5A129557  Ack: 0x53A7A3BA  Win: 0x4038  TcpLen: 20

[**] [116:46:1] (snort_decoder) WARNING: TCP Data Offset is less than 5! [**]
[Classification: Potential Corporate Privacy Violation] [Priority: 1]
10/10-08:53:55.446507 62.13.27.29:0 -> 32.245.113.30:0
TCP TTL:234 TOS:0x0 ID:0 IpLen:20 DgmLen:40
*****R** Seq: 0x1EC00EE  Ack: 0x1EC00EE  Win: 0x0  TcpLen: 0

[**] [1:620:6] SCAN Proxy Port 8080 attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-16:00:56.996507 66.159.18.66:43517 -> 170.129.50.120:8080
TCP TTL:53 TOS:0x0 ID:59575 IpLen:20 DgmLen:60 DF
******S* Seq: 0xBED8745  Ack: 0x0  Win: 0x16D0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 48656370 0 NOP WS: 0

[**] [116:46:1] (snort_decoder) WARNING: TCP Data Offset is less than 5! [**]
[Classification: Potential Corporate Privacy Violation] [Priority: 1]
11/09-20:51:11.676507 62.13.27.29:0 -> 207.166.33.145:0
TCP TTL:234 TOS:0x0 ID:0 IpLen:20 DgmLen:40
*****R** Seq: 0x81F9750  Ack: 0x81F9750  Win: 0x0  TcpLen: 0

[**] [116:46:1] (snort_decoder) WARNING: TCP Data Offset is less than 5! [**]
[Classification: Potential Corporate Privacy Violation] [Priority: 1]
11/09-20:51:11.676507 62.13.27.29:0 -> 207.166.33.145:0
TCP TTL:234 TOS:0x0 ID:0 IpLen:20 DgmLen:40
*****R** Seq: 0x81F9750  Ack: 0x81F9750  Win: 0x0  TcpLen: 0

[**] [116:46:1] (snort_decoder) WARNING: TCP Data Offset is less than 5! [**]
[Classification: Potential Corporate Privacy Violation] [Priority: 1]
11/12-20:25:11.826507 217.209.183.235:0 -> 207.166.252.249:0
TCP TTL:236 TOS:0x0 ID:0 IpLen:20 DgmLen:40
*****R** Seq: 0x24A1C4C  Ack: 0x24A1C4C  Win: 0x0  TcpLen: 0

[**] [116:46:1] (snort_decoder) WARNING: TCP Data Offset is less than 5! [**]
[Classification: Potential Corporate Privacy Violation] [Priority: 1]
11/12-20:25:11.826507 217.209.183.235:0 -> 207.166.252.249:0
TCP TTL:236 TOS:0x0 ID:0 IpLen:20 DgmLen:40
*****R** Seq: 0x24A1C4C  Ack: 0x24A1C4C  Win: 0x0  TcpLen: 0

[**] [1:615:5] SCAN SOCKS Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-16:00:56.996507 66.159.18.66:43520 -> 170.129.50.120:1080
TCP TTL:53 TOS:0x0 ID:4253 IpLen:20 DgmLen:60 DF
******S* Seq: 0xB9A7F6F  Ack: 0x0  Win: 0x16D0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 48656370 0 NOP WS: 0
[Xref => http://help.undernet.org/proxyscan/]

[**] [1:615:5] SCAN SOCKS Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-16:00:56.996507 66.159.18.66:43520 -> 170.129.50.120:1080
TCP TTL:53 TOS:0x0 ID:4253 IpLen:20 DgmLen:60 DF
******S* Seq: 0xB9A7F6F  Ack: 0x0  Win: 0x16D0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 48656370 0 NOP WS: 0
```

```
[Xref => http://help.undernet.org/proxyscan/]

[**] [1:524:7] BAD-TRAFFIC tcp port 0 traffic [**]
[Classification: Misc activity] [Priority: 3]
11/15-07:35:25.406507 211.47.255.24:41358 -> 170.129.195.40:0
TCP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0xD4C4A857  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

[**] [1:524:7] BAD-TRAFFIC tcp port 0 traffic [**]
[Classification: Misc activity] [Priority: 3]
11/15-07:35:31.326507 211.47.255.24:41358 -> 170.129.195.40:0
TCP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0xD4C4A857  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

[**] [1:524:7] BAD-TRAFFIC tcp port 0 traffic [**]
[Classification: Misc activity] [Priority: 3]
11/15-07:35:57.316507 211.47.255.24:41611 -> 170.129.195.40:0
TCP TTL:46 TOS:0x0 ID:0 IpLen:20 DgmLen:52 DF
******S* Seq: 0xD60E9D41  Ack: 0x0  Win: 0x16D0  TcpLen: 32
TCP Options (6) => MSS: 1460 NOP NOP SackOK NOP WS: 0

[**] [1:615:5] SCAN SOCKS Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/14-16:00:56.996507 66.159.18.66:43521 -> 170.129.50.120:1080
TCP TTL:53 TOS:0x0 ID:2662 IpLen:20 DgmLen:60 DF
******S* Seq: 0xBF46594  Ack: 0x0  Win: 0x16D0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 48656370 0 NOP WS: 0
[Xref => http://help.undernet.org/proxyscan/]

Cluster 17: (2.3879)
--------------------
[**] [1:620:6] SCAN Proxy Port 8080 attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
11/18-13:57:27.629436 10.10.10.212:3405 -> 172.20.11.238:8080
TCP TTL:128 TOS:0x0 ID:24091 IpLen:20 DgmLen:48 DF
******S* Seq: 0xCDD85F65  Ack: 0x0  Win: 0xFC00  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

Cluster 18: (2.4082)
--------------------
[**] [119:13:1] (http_inspect) NON-RFC HTTP DELIMITER [**]
11/15-10:00:30.986507 170.129.50.120:64645 -> 199.45.45.132:80
TCP TTL:125 TOS:0x0 ID:1655 IpLen:20 DgmLen:829 DF
***AP*** Seq: 0x68E6FA3B  Ack: 0x43F74153  Win: 0x4470  TcpLen: 20

Cluster 19: (2.4340)
--------------------
[**] [116:46:1] (snort_decoder) WARNING: TCP Data Offset is less than 5! [**]
[Classification: Potential Corporate Privacy Violation] [Priority: 1]
09/05-07:34:31.984488 172.20.10.199:0 -> 138.97.150.9:0
TCP TTL:237 TOS:0x0 ID:0 IpLen:20 DgmLen:40
*****R** Seq: 0x58092C9A  Ack: 0x58092C9A  Win: 0x0  TcpLen: 12

Cluster 20: (2.4597)
--------------------
[**] [116:46:1] (snort_decoder) WARNING: TCP Data Offset is less than 5! [**]
[Classification: Potential Corporate Privacy Violation] [Priority: 1]
09/05-08:35:49.004488 172.20.10.199:0 -> 138.97.54.152:0
TCP TTL:236 TOS:0x0 ID:0 IpLen:20 DgmLen:40
*****R** Seq: 0x5841481E  Ack: 0x5841481E  Win: 0x0  TcpLen: 4

[**] [119:4:1] (http_inspect) BARE BYTE UNICODE ENCODING [**]
```

```
11/15-04:45:11.546507 170.129.50.120:61044 -> 216.136.173.111:80
TCP TTL:125 TOS:0x0 ID:9584 IpLen:20 DgmLen:932 DF
***AP*** Seq: 0x9F68995  Ack: 0x6193010  Win: 0xFAF0  TcpLen: 20

[**] [119:12:1] (http_inspect) APACHE WHITESPACE (TAB) [**]
11/15-04:45:21.306507 170.129.50.120:61044 -> 216.136.173.111:80
TCP TTL:125 TOS:0x0 ID:10054 IpLen:20 DgmLen:932 DF
***AP*** Seq: 0x9FEC995  Ack: 0x6193010  Win: 0xFAF0  TcpLen: 20

[**] [116:46:1] (snort_decoder) WARNING: TCP Data Offset is less than 5! [**]
[Classification: Potential Corporate Privacy Violation] [Priority: 1]
11/10-01:28:34.556507 62.13.27.29:0 -> 207.166.78.44:0
TCP TTL:234 TOS:0x0 ID:0 IpLen:20 DgmLen:40 DF
*****R** Seq: 0x91D8C02  Ack: 0x91D8C02  Win: 0x0  TcpLen: 12

[**] [119:4:1] (http_inspect) BARE BYTE UNICODE ENCODING [**]
11/14-09:54:38.316507 170.129.50.120:63362 -> 159.153.199.24:80
TCP TTL:125 TOS:0x0 ID:38908 IpLen:20 DgmLen:436 DF
***AP*** Seq: 0x99AD8FC7  Ack: 0x732FBFCD  Win: 0x4230  TcpLen: 20

[**] [119:4:1] (http_inspect) BARE BYTE UNICODE ENCODING [**]
11/14-09:54:47.286507 170.129.50.120:63387 -> 159.153.199.24:80
TCP TTL:125 TOS:0x0 ID:38984 IpLen:20 DgmLen:436 DF
***AP*** Seq: 0x99C8B003  Ack: 0x733D8EE2  Win: 0x4230  TcpLen: 20

[**] [119:4:1] (http_inspect) BARE BYTE UNICODE ENCODING [**]
11/14-09:54:47.286507 170.129.50.120:63387 -> 159.153.199.24:80
TCP TTL:125 TOS:0x0 ID:38984 IpLen:20 DgmLen:436 DF
***AP*** Seq: 0x99C8B003  Ack: 0x733D8EE2  Win: 0x4230  TcpLen: 20

Cluster 21: (2.5271)
--------------------
[**] [116:46:1] (snort_decoder) WARNING: TCP Data Offset is less than 5! [**]
[Classification: Potential Corporate Privacy Violation] [Priority: 1]
09/29-14:13:00.736507 210.134.225.210:0 -> 115.74.203.32:0
TCP TTL:44 TOS:0x0 ID:36567 IpLen:20 DgmLen:60 DF
*2*A**S* Seq: 0x2D45F8DA  Ack: 0x0  Win: 0xA622  TcpLen: 16

[**] [116:46:1] (snort_decoder) WARNING: TCP Data Offset is less than 5! [**]
[Classification: Potential Corporate Privacy Violation] [Priority: 1]
10/10-15:12:45.926507 80.128.206.166:0 -> 32.245.166.119:0
TCP TTL:117 TOS:0x0 ID:25125 IpLen:20 DgmLen:40 DF
*******F Seq: 0x720049  Ack: 0xD655185A  Win: 0x5010  TcpLen: 0

Cluster 22: (2.5561)
--------------------
[**] [119:4:1] (http_inspect) BARE BYTE UNICODE ENCODING [**]
11/14-13:03:42.666507 170.129.50.120:63598 -> 64.4.22.250:80
TCP TTL:124 TOS:0x0 ID:56064 IpLen:20 DgmLen:932 DF
***AP*** Seq: 0x94851318  Ack: 0x9AB18054  Win: 0x43E1  TcpLen: 20

Cluster 23: (2.5994)
--------------------
[**] [1:648:6] SHELLCODE x86 NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
11/14-21:55:36.676507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0x0 ID:46498 IpLen:20 DgmLen:1420 DF
***A**** Seq: 0xA0750D60  Ack: 0x90CF9E29  Win: 0x16D0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS181]

[**] [119:4:1] (http_inspect) BARE BYTE UNICODE ENCODING [**]
11/15-02:46:26.726507 170.129.50.120:64749 -> 216.130.211.11:80
TCP TTL:124 TOS:0x0 ID:26873 IpLen:20 DgmLen:1332 DF
***AP*** Seq: 0x1600D507  Ack: 0xF2FBCCD5  Win: 0x2058  TcpLen: 20
```

```
[**] [119:12:1] (http_inspect) APACHE WHITESPACE (TAB) [**]
11/15-02:46:28.446507 170.129.50.120:64749 -> 216.130.211.11:80
TCP TTL:124 TOS:0x0 ID:40697 IpLen:20 DgmLen:1332 DF
***AP*** Seq: 0x1601F507  Ack: 0xF2FBCCD5  Win: 0x2058  TcpLen: 20

Cluster 24: (2.6599)
--------------------
[**] [1:523:4] BAD-TRAFFIC ip reserved bit set [**]
[Classification: Misc activity] [Priority: 3]
11/15-01:09:54.346507 200.200.200.1 -> 170.129.217.111
TCP TTL:241 TOS:0x0 ID:0 IpLen:20 DgmLen:40 RB
Frag Offset: 0x0864   Frag Size: 0x0014

[**] [1:523:4] BAD-TRAFFIC ip reserved bit set [**]
[Classification: Misc activity] [Priority: 3]
11/15-01:21:09.846507 200.200.200.1 -> 170.129.127.227
TCP TTL:241 TOS:0x0 ID:0 IpLen:20 DgmLen:40 RB
Frag Offset: 0x0864   Frag Size: 0x0014

[**] [1:523:4] BAD-TRAFFIC ip reserved bit set [**]
[Classification: Misc activity] [Priority: 3]
11/14-11:21:09.916507 200.200.200.1 -> 170.129.211.200
TCP TTL:242 TOS:0x0 ID:0 IpLen:20 DgmLen:40 RB
Frag Offset: 0x0864   Frag Size: 0x0014

[**] [1:523:4] BAD-TRAFFIC ip reserved bit set [**]
[Classification: Misc activity] [Priority: 3]
11/14-11:21:09.916507 200.200.200.1 -> 170.129.211.200
TCP TTL:242 TOS:0x0 ID:0 IpLen:20 DgmLen:40 RB
Frag Offset: 0x0864   Frag Size: 0x0014

[**] [1:523:4] BAD-TRAFFIC ip reserved bit set [**]
[Classification: Misc activity] [Priority: 3]
11/14-14:37:18.296507 200.200.200.1 -> 170.129.2.16
TCP TTL:242 TOS:0x0 ID:0 IpLen:20 DgmLen:40 RB
Frag Offset: 0x0864   Frag Size: 0x0014

[**] [1:523:4] BAD-TRAFFIC ip reserved bit set [**]
[Classification: Misc activity] [Priority: 3]
11/14-15:54:39.456507 200.200.200.1 -> 170.129.79.180
TCP TTL:242 TOS:0x0 ID:0 IpLen:20 DgmLen:40 RB
Frag Offset: 0x0864   Frag Size: 0x0014

[**] [1:523:4] BAD-TRAFFIC ip reserved bit set [**]
[Classification: Misc activity] [Priority: 3]
11/14-17:59:31.346507 200.200.200.1 -> 170.129.239.44
TCP TTL:242 TOS:0x0 ID:0 IpLen:20 DgmLen:40 RB
Frag Offset: 0x0864   Frag Size: 0x0014

Cluster 25: (2.6824)
--------------------
[**] [116:46:1] (snort_decoder) WARNING: TCP Data Offset is less than 5! [**]
[Classification: Potential Corporate Privacy Violation] [Priority: 1]
11/09-10:21:40.336507 172.20.10.199:0 -> 207.166.124.204:0
TCP TTL:235 TOS:0x0 ID:0 IpLen:20 DgmLen:40
*****R** Seq: 0xBAA36CBC  Ack: 0xBAA36CBC  Win: 0x0  TcpLen: 8

[**] [116:46:1] (snort_decoder) WARNING: TCP Data Offset is less than 5! [**]
[Classification: Potential Corporate Privacy Violation] [Priority: 1]
11/10-01:12:17.866507 172.20.10.199:0 -> 207.166.119.62:0
TCP TTL:235 TOS:0x0 ID:0 IpLen:20 DgmLen:40
*****R** Seq: 0xBDD2D468  Ack: 0xBDD2D468  Win: 0x0  TcpLen: 16
```

```
[**] [116:46:1] (snort_decoder) WARNING: TCP Data Offset is less than 5! [**]
[Classification: Potential Corporate Privacy Violation] [Priority: 1]
11/10-01:12:17.866507 172.20.10.199:0 -> 207.166.119.62:0
TCP TTL:235 TOS:0x0 ID:0 IpLen:20 DgmLen:40
*****R** Seq: 0xBDD2D468  Ack: 0xBDD2D468  Win: 0x0  TcpLen: 16

[**] [116:46:1] (snort_decoder) WARNING: TCP Data Offset is less than 5! [**]
[Classification: Potential Corporate Privacy Violation] [Priority: 1]
11/09-12:39:54.816507 172.20.10.199:0 -> 207.166.114.79:0
TCP TTL:234 TOS:0x0 ID:0 IpLen:20 DgmLen:40
*****R** Seq: 0xBB21FD8E  Ack: 0xBB21FD8E  Win: 0x0  TcpLen: 4

Cluster 26: (2.7032)
--------------------
[**] [116:46:1] (snort_decoder) WARNING: TCP Data Offset is less than 5! [**]
[Classification: Potential Corporate Privacy Violation] [Priority: 1]
11/11-09:08:23.926507 172.20.10.199:0 -> 207.166.207.98:0
TCP TTL:234 TOS:0x0 ID:0 IpLen:20 DgmLen:40
*****R** Seq: 0xC4AD19A6  Ack: 0xC4AD19A6  Win: 0x0  TcpLen: 16

[**] [116:46:1] (snort_decoder) WARNING: TCP Data Offset is less than 5! [**]
[Classification: Potential Corporate Privacy Violation] [Priority: 1]
11/10-07:20:11.976507 172.20.10.199:0 -> 207.166.168.10:0
TCP TTL:235 TOS:0x0 ID:0 IpLen:20 DgmLen:40
*****R** Seq: 0xBF23A8C4  Ack: 0xBF23A8C4  Win: 0x0  TcpLen: 0

Cluster 27: (2.7592)
--------------------
[**] [1:1390:4] SHELLCODE x86 inc ebx NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
11/14-16:10:30.806507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:56986 IpLen:20 DgmLen:1420 DF
***A**** Seq: 0x8217BFFC  Ack: 0x90CF9E29  Win: 0x16D0  TcpLen: 20

[**] [1:1390:4] SHELLCODE x86 inc ebx NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
11/14-16:10:30.806507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:56986 IpLen:20 DgmLen:1420 DF
***A**** Seq: 0x8217BFFC  Ack: 0x90CF9E29  Win: 0x16D0  TcpLen: 20

[**] [1:648:6] SHELLCODE x86 NOOP [**]
[Classification: Executable code was detected] [Priority: 1]
11/14-21:55:36.566507 129.118.2.10:57425 -> 170.129.50.120:63414
TCP TTL:51 TOS:0xA0 ID:46490 IpLen:20 DgmLen:1420 DF
***A**** Seq: 0xA074E240  Ack: 0x90CF9E29  Win: 0x16D0  TcpLen: 20
[Xref => http://www.whitehats.com/info/IDS181]

Cluster 28: (2.8017)
--------------------
[**] [116:46:1] (snort_decoder) WARNING: TCP Data Offset is less than 5! [**]
[Classification: Potential Corporate Privacy Violation] [Priority: 1]
09/26-11:25:29.286507 209.10.208.246:0 -> 115.74.249.65:0
TCP TTL:96 TOS:0x58 ID:46247 IpLen:20 DgmLen:40 DF
***A*R** Seq: 0xEEAA80A9  Ack: 0xB3BBCFD  Win: 0x0  TcpLen: 0

Cluster 29: (2.8412)
--------------------
[**] [1:2314:1] SHELLCODE x86 0x90 NOOP unicode [**]
[Classification: Executable code was detected] [Priority: 1]
11/15-12:09:01.256507 216.183.64.22:43124 -> 170.129.50.3:20
TCP TTL:235 TOS:0x0 ID:42290 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0xAC0E4063  Ack: 0x9ED7097C  Win: 0x65D0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 208895639 6108072
```

```
Cluster 30: (2.9358)
--------------------
[**] [116:46:1] (snort_decoder) WARNING: TCP Data Offset is less than 5! [**]
[Classification: Potential Corporate Privacy Violation] [Priority: 1]
11/12-18:38:17.846507 203.80.239.162:0 -> 207.166.182.137:0
TCP TTL:107 TOS:0x0 ID:35119 IpLen:20 DgmLen:48 DF
1*UA**** Seq: 0x7930005  Ack: 0xD80A04D1  Win: 0x64BA  TcpLen: 0  UrgPtr: 0x800

[**] [116:46:1] (snort_decoder) WARNING: TCP Data Offset is less than 5! [**]
[Classification: Potential Corporate Privacy Violation] [Priority: 1]
11/12-18:38:17.846507 203.80.239.162:0 -> 207.166.182.137:0
TCP TTL:107 TOS:0x0 ID:35119 IpLen:20 DgmLen:48 DF
1*UA**** Seq: 0x7930005  Ack: 0xD80A04D1  Win: 0x64BA  TcpLen: 0  UrgPtr: 0x800

Cluster 31: (3.0080)
--------------------
[**] [116:47:1] (snort_decoder) WARNING: TCP Header length exceeds packet length! [**]
[Classification: Potential Corporate Privacy Violation] [Priority: 1]
09/29-14:30:07.986507 210.134.225.210:0 -> 115.74.100.16:0
TCP TTL:44 TOS:0x0 ID:29204 IpLen:20 DgmLen:60 DF
12*A*R*F Seq: 0x4C52A622  Ack: 0x29603  Win: 0x800  TcpLen: 60

Cluster 32: (3.0448)
--------------------
[**] [116:47:1] (snort_decoder) WARNING: TCP Header length exceeds packet length! [**]
[Classification: Potential Corporate Privacy Violation] [Priority: 1]
09/29-14:22:48.726507 210.134.225.210:0 -> 115.74.35.151:0
TCP TTL:44 TOS:0x0 ID:10754 IpLen:20 DgmLen:60 DF
12*A*R*F Seq: 0x4C52A622  Ack: 0x29603  Win: 0x800  TcpLen: 60

[**] [116:47:1] (snort_decoder) WARNING: TCP Header length exceeds packet length! [**]
[Classification: Potential Corporate Privacy Violation] [Priority: 1]
09/29-14:21:12.706507 210.134.225.210:0 -> 115.74.21.137:0
TCP TTL:44 TOS:0x0 ID:7883 IpLen:20 DgmLen:60 DF
12*A*R*F Seq: 0x4C52A622  Ack: 0x29603  Win: 0x800  TcpLen: 60
```

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.) School of Information Technology and Engineering University of Ottawa Ottawa Ontario	2. SECURITY CLASSIFICATION (overall security classification of the document including special warning terms if applicable). UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C,R or U) in parentheses after the title).

Autocorrel I: A Neural Network Based Network Event Correlation Approach

4. AUTHORS
(Last name, first name, middle initial. If military, show rank, e.g. Doe, Maj. John E.)

Japkowicz, Nathalie ; Smith, Reuben

5. DATE OF PUBLICATION (month and year of publication of document) May 2005	6a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc). 152	6b. NO. OF REFS (total cited in document) 34

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered).

Contractor Report

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include address).

DEFENCE R&D CANADA - OTTAWA
3701 Carling Avenue, Ottawa, Ontario, K1A 0Z4

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Specify whether project or grant). 15BF29	9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written). W7714-3-08710
10a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique.) DRDC Ottawa CR 2005-030	10b. OTHER DOCUMENT NOs. (Any other numbers which may be assigned this document either by the originator or by the sponsor.)

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(X) Unlimited distribution
(   ) Defence departments and defence contractors; further distribution only as approved
(   ) Defence departments and Canadian defence contractors; further distribution only as approved
(   ) Government departments and agencies; further distribution only as approved
(   ) Defence departments; further distribution only as approved
(   ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution beyond the audience specified in (11) is possible, a wider announcement audience may be selected).

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

Network event correlation is the process where correlations between network events are discovered and reported. Network intrusion detection analysts who have capable event correlation software at their disposal are more effective because the software can give an intrusion analyst a broader view of the threats posed to their system. The event correlation information is used by a network administrator to deduce the true relationship between individual network events. The autoassociator is ideally suited to the task of network event correlation. The autoassociator is a specialized piece of neural network architecture that can be used to cluster numerically similar data instances. We use the autoassociator to build prototype software to cluster network alerts generated by a Snort intrusion detection system, and discuss how the results are significant, and how they can be applied to other types of network events.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title).

Neural Network, Intrusion Detection System, Network Event Correlation, Autoassociator

DRDC Ottawa DOCUMENT SUBMISSION RECORD

Revised 02/05/27

Work Unit # 15BF29 DRP # CR 2005-030

This form is to accompany all documents to be reviewed by the Document Review Panel (DRP)

TITLE & TITLE CLASSIFICATION Enter The title followed by classification in the form (U) for UNCLASSIFIED, (C) for CONFIDENTIAL, (S) for SECRET, and (TS) for TOP SECRET. For classified titles, page classification must also be changed in Header/Footer.
Autocorrel I: A Neural Network Based Network Event Correlation Approach (U)

AUTHOR(S): Japkowicz, Nathalie ; Smith, Reuben

DOCUMENT CLASSIFICATION/WARNING TERMS: (U)

RESPONSIBLE SECTION: INFORMATION OPERATIONS

DOCUMENT INTENDED FOR:

☐ DRDC Ottawa TECHNICAL REPORT
☐ DRDC Ottawa TECHNICAL MEMORANDUM
☐ DRDC Ottawa TECHNICAL NOTE (for dist. outside DRDC Ottawa)
☐ PAPER FOR OPEN LITERATURE
☒ CONTRACTOR REPORT (Name of Company)
School of Information Technology and Engineering

☐ CRC REPORT
☐ CRC TECHNICAL NOTE
☐ ABSTRACT
☐ ORAL PRESENTATION
☐ WORKING PAPER FOR TTCP
☐ WORKING PAPER FOR NATO

Use this space to identify the following:

1. Distribution of document to any agency not included on Section Standard Distribution List (name agency)
2. Name the publication to be printed in or location of the presentation Date

Each signature identifies that the attached document, forms, and distribution list are accurate and complete and ready for the next stage of review.

AUTHOR/SA: ____________________ **DATE** __________
REVIEWER: ____________________ **DATE** __________
EDITOR: ____________________ **DATE** __________
SECTION HEAD: ____________________ **DATE** __________
DRP REP: ____________________ **DATE** __________
DRP CHAIR: ____________________ **DATE** __________